CHỦ BIÊN: BÙI CÔNG CƯỜNG
NGUYỄN DOÃN PHƯỚC

HỆ MỜ MẠNG NƠRON & ỨNG DỤNG



NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT

Chủ biên: Bùi Công Cường
Nguyễn Doãn Phước

HỆ MỜ, MẠNG NƠRON VÀ ỨNG DỤNG

(TUYỂN TẬP CÁC BÀI GIẢNG)

IN LẦN THỨ HAI, CÓ SỬA ĐỔI VÀ BỔ SUNG



NHÀ XUẤT BẢN KHOA HỌC VÀ KỸ THUẬT
– 2006 –

Chịu trách nhiệm xuất bản: PGS.TS **Tô Đăng Hải**
Biên tập: **Nguyễn Đăng**

Vẽ bìa: **Ngọc Quang**

In 800 cuốn, khổ 16 x 24cm, tại Xưởng in NXB Văn hoá Dân tộc
Quyết định xuất bản số: 409-2006/CXB/2-33/KHKT
In xong và nộp lưu chiểu Quý IV năm 2006.

Lời nói đầu

Từ 20 năm nay, Lý thuyết tập mờ và Mạng nơ-ron nhân tạo đã phát triển rất nhanh và đa dạng. Công nghệ mờ và công nghệ mạng nơ-ron đã cung cấp những công nghệ mới cho các ngành công nghiệp làm ra nhiều sản phẩm thông minh, đáp ứng nhu cầu thị trường cần có những bộ điều khiển linh hoạt hơn, những thiết bị "biết" làm việc với những bài toán khó, phải xử lý nhiều loại thông tin mập mờ, chưa đầy đủ và thiếu chính xác. Hai công nghệ hiện đại này là hai trụ cột chính để tạo nên công nghệ tích hợp mới, công nghệ tính toán mềm (soft computing).

Để đáp ứng nhu cầu mang công nghệ mới vào nước ta và trực tiếp cung cấp cho sinh viên và các cán bộ kỹ thuật trẻ những kiến thức cơ bản nhất về lĩnh vực này, Trường Thu về "Hệ mờ và ứng dụng" lần thứ nhất đã tổ chức tại Hà nội, tháng 8/2000. Từ những bài giảng đã trình bày tại Trường Thu, chúng tôi chọn lọc, nâng cấp và bổ sung thành những chương khá hoàn chỉnh của cuốn sách này.

Cuốn sách bắt đầu với hai chương tổng quan do Bùi Công Cường và Nguyễn Cát Hồ viết. Nếu chương đầu tập trung vào những kiến thức cơ bản của hai trụ cột chính: Hệ mờ và Mạng nơ-ron nhân tạo, thì trong chương hai sau những phần về toán học mờ, quy hoạch mờ, tác giả đã tập trung trình bày khá hệ thống những vấn đề rất cơ bản thuộc một tên gọi chung "Công nghệ tính toán mềm".

Tiếp theo là những chương chuyên sâu hơn, như Logic mờ và các ứng dụng đa dạng, Điều khiển mờ và mạng nơ-ron của Nguyễn Doãn Phước và Phan Xuân Minh. Lý thuyết khả năng – một hướng hiện đại nằm giữa Lý thuyết tập mờ và Lý thuyết xác suất – do Đỗ Văn Thành trình bày.

Tiếp theo là bài giảng của tập thể Nguyễn Thanh Thuỷ, Nguyễn Hữu Đức và Trần Ngọc Hà – một bài giảng rất hay về tích hợp các kỹ thuật tính toán mềm và mạng nơ-ron trong xử lý dữ liệu và bài giảng về một lớp toán tử gộp mới – toán tử trung bình trọng số có sắp xếp. Chắc chắn các dạng suy rộng của nó chứa đựng nhiều khả năng phát triển và ứng dụng.

Ba chương cuối của cuốn sách tập trung vào lĩnh vực hiện đại: Mạng nơ-ron nhân tạo và ứng dụng. Nếu như bài giảng của Vũ Như Lân tập trung vào hai bài toán chính, khó và rất hay của Điều khiển học kỹ thuật: Nhận dạng mô hình và điều khiển các hệ thống phi tuyến, thì bài giảng của Đặng Quang Á lại tập trung trong một số lớp thuật toán giải các bài toán tối ưu rời rạc.

Cuối cùng cần nhắc tới bài giảng có liên quan tới dự báo. Trong khuôn khổ của "Công nghệ tính toán mềm", kết hợp Giải thuật di truyền và mạng nơ-ron để dự báo đó là một hướng hiện đại và đầy triển vọng – vấn đề này được đề cập đến trong bài giảng của Nguyễn Thanh Thuỷ và Nguyễn Thị Diệu Thư.

Sẽ là thiếu sót nếu không kể đến một đặc thù của cuốn sách. Sau rất nhiều chương có phần tài liệu dẫn khá phong phú, đủ kiến thức để các bạn đọc có thể đi sâu tiếp. Hơn nữa theo chỗ chúng tôi biết thì các tài liệu dẫn này hiện có trong tay các tác giả.

Không còn nghi ngờ gì nữa, hơn mười bài giảng trên đã tạo một bó hoa khá hoàn chỉnh, đa sắc, nhiều thông tin, đưa tới cho bạn đọc những kiến thức rất cơ bản đồng thời gợi mở cho các bạn sinh viên trẻ nhiều hướng nghiên cứu triển vọng và đầy hấp dẫn.

Cuốn sách sẽ không thể ra mắt bạn đọc nếu không có sự hợp tác nhiệt tình của các tác giả, nếu không có sự đỡ đầu chủ yếu của Viện Toán học – đơn vị đóng góp chính tổ chức Trường Thu về "Hệ mờ và ứng dụng", nếu không có sự giúp đỡ và góp ý quý báu của Ban Biên tập Nhà Xuất bản Khoa học và Kỹ thuật. Với tất cả các cá nhân và đơn vị trên chúng tôi xin chân thành cám ơn.

Do nhiều hạn chế, đặc biệt hạn chế về thời gian, cuốn sách không tránh khỏi những khiếm khuyết. Chúng tôi hoan nghênh và chân thành lắng nghe mọi góp ý.

Hà Nội, ngày 6.3.2006

Mục lục

1 KIẾN THỨC CƠ SỞ CỦA HỆ MỜ

Bùi Công Cường
Viện Toán học Hà nội

1 Tập mờ, logic mờ và hệ mờ

1.1 Tập mờ

Trong phần 1 của chương này chúng ta bắt đầu tìm hiểu những khái niệm cơ bản nhất: định nghĩa tập mờ của L.Zadeh (1965), các phép toán đại số, nguyên lý suy rộng, số mờ và sau đó là khái niệm biến ngôn ngữ, các phép toán bước đầu của logic mờ, suy diễn mờ và hệ mờ trên cơ sở các luật mờ. Một số dạng phát triển ứng dụng quan trọng cũng sẽ được lướt nhanh để tạo điều kiện cho các bạn mới học lần đầu có cái nhìn tổng quan về hệ mờ và ứng dụng.

1.1.1 Định nghĩa tập mờ

Xét tập $X \neq \emptyset$. Ta sẽ gọi X là không gian nền. Chẳng hạn:

X = tập sinh viên Đại học Bách khoa Hà Nội khoá 41.

$A1$ = tập sinh viên Khoa Công nghệ thông tin khoá 41.

Khi đó $A1$ là một *tập con rõ* của X. Gọi:

$A2$ = tập sinh viên giỏi Tin , khoá 41 của Khoa Cơ khí.

Khi đó $A2$ là một *tập mờ* trên X.

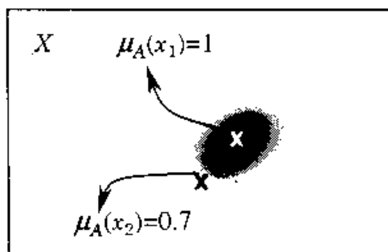
Một minh hoạ khác về tập mờ là vết vân tay của tội phạm để lại trên hiện trường.

Định nghĩa 1.1 (xem[1]): A là tập mờ trên không gian nền X nếu A được xác định bởi hàm

$\mu_A : X \rightarrow [0,1]$.

μ_A là hàm thuộc (*membership function*) còn $\mu_A(x)$ là độ thuộc của x vào tập mờ A.

Ví dụ 1.1



Hình 1 : Ví dụ về tập mờ

Nhiều tài liệu vẫn quen ký hiệu $\mu_A(x)$. Tuy nhiên, để gọn đôi khi cần ta sẽ ký hiệu $A(x)$ thay cho $\mu_A(x)$.

Chúng ta cũng sẽ kí hiệu

$$A = \{(\mu_A(x)/x): x \in X\}.$$

Ví dụ 1.2: A_0 = một vài (quả cam) = { (0/0) , (0/1) . (0.6/2) . (1/3) . (1/4) . (0.8/5) . (0.2/6) }.

Ví dụ 1.3: A = "số thực gần 10" có hàm thuộc $\mu_A(x) = \dfrac{1}{1+(x-10)^2}$.

Ta sẽ kí hiệu

$$F(X) = \{A \text{ tập mờ trên } X\}$$

Định nghĩa 1.2: Giá của tập mờ A, $S(A)$ là tập các điểm x nào có $\mu_A(x)>0$.

Với mỗi $0 \le \alpha \le 1$ tập mức A_α cho bởi:

$$A_\alpha = \{x \in X : \mu_A(x) \ge \alpha\}.$$

Để ý A_α là tập con rõ của X.

Mệnh đề 1.1: Cho A là tập mờ. Khi đó:

$$\mu_A(x) = \sup_{0\le\alpha\le1} \min(\alpha, \mu_{A_\alpha}(x)), \quad \text{với} \quad \mu_{A_\alpha}(x) = \begin{cases} 1 & \text{nếu } x \in A_\alpha \\ 0 & \text{nếu } x \notin A_\alpha \end{cases}$$

(Ở đây sup- là cận trên đúng của một tập trên đường thẳng số thực. Bạn nào chưa quen có thể thay bằng max, hoặc hỏi thêm thầy dạy Toán).

Chứng minh: Cho $0 < \alpha \le 1$ cố định. Với x có $\mu_A(x) = 0$. Do $x \notin A_\alpha$, nên $\mu_{A_\alpha}(x) = 0$.

Vậy: $\sup\limits_{0\le\alpha\le1} \min(\alpha, \mu_{A_\alpha}(x)) = 0 = \mu_A(x)$.

Với x có $\mu_A(x) = \alpha' > 0$. Ta xét 3 trường hợp:

– Nếu $\alpha < \alpha'$, $\mu_{A_\alpha}(x) = 1$, nên $\min(\alpha, \mu_{A_\alpha}(x)) = \alpha < \alpha'$.

– Nếu $\alpha = \alpha'$, $\mu_{A_\alpha}(x) = 1$, nên $\min(\alpha, \mu_{A_\alpha}(x)) = \alpha = \alpha'$.

– Nếu $\alpha > \alpha'$, $\mu_{A_\alpha}(x) = 0$, nên $\min(\alpha, \mu_{A_\alpha}(x)) = 0'$.

Vậy: $\sup\limits_{0\le\alpha\le1} \min(\alpha, \mu_{A_\alpha}(x)) = \alpha' = \mu_A(x)$.

1.1.2 Các phép toán đại số trên tập mờ

Định nghĩa 1.3: Cho A, B là hai tập mờ trên không gian nền X, có các hàm thuộc μ_A, μ_B. Khi đó phép hợp $A \cup B$, phép giao $A \cap B$ và phần bù A^C là các tập mờ trên X với các hàm thuộc cho bởi:

$$\mu_{A\cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}, \quad x \in X$$

$$\mu_{A\cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}, \quad x \in X$$

$$\mu_{A^C}(x) = 1 - \mu_A(x), \quad x \in X.$$

Định nghĩa 1.4: Cho $A, B \in F(X)$. Ta nói:

$A \subseteq B$, nếu $\mu_A(x) \le \mu_B(x)$ với mọi $x \in X$.

$A \supseteq B$, nếu $\mu_A(x) \ge \mu_B(x)$ với mọi $x \in X$.

Do đó

$A = B$, nếu $\mu_A(x) = \mu_B(x)$ với mọi $x \in X$.

Dễ dàng kiểm tra mệnh đề sau:

Mệnh đề 1.2: Cho $A, B \in F(X)$. Khi đó

$(A \cup B)_\alpha = A_\alpha \cup B_\alpha$ và $(A \cap B)_\alpha = A_\alpha \cap B_\alpha$.

Ta sẽ coi $\varnothing$ là tập mờ với $\mu_\varnothing(x) = 0$ với mọi x. X là tập mờ với $\mu_X(x) = 1$, với mọi x.

Với các tập mờ nhiều tính chất của tập rõ còn đúng. Mệnh đề sau sẽ minh họa điều đó.

Mệnh đề 1.3: Cho $A, B, C \in F(X)$. Ta có các tính chất:

a) Giao hoán: $A \cup B = B \cup A$, $A \cap B = B \cap A$

b) Kết hợp: $A \cup (B \cup C) = A \cup (B \cup C)$

$A \cap (B \cap C) = A \cap (B \cap C)$

c) Luỹ đẳng: $A \cup A = A$, $A \cap A = A$

d) Phân phối: $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

e) $A \cap \varnothing = \varnothing$ và $A \cup X = X$

f) Đồng nhất: $A \cup \varnothing = A$ và $A \cap X = A$

g) Hấp thu $A \cup (A \cap B) = A$ và $A \cap (A \cup B) = A$

h) Luật De Morgan $(A \cup B)^C = A^C \cap B^C$ và $(A \cap B)^C = A^C \cup B^C$

i) Cuộn: $(A^C)^C = A$

j) Dạng tương đương: $(A^C \cup B) \cap (A \cup B^C) = (A^C \cap B^C) \cup (A \cap B)$

k) Hiệu đối xứng: $(A^C \cap B) \cup (A \cap B^C) = (A^C \cup B^C) \cap (A \cup B)$

Chứng minh. Ta sẽ chỉ chứng minh một vài đẳng thức để minh hoạ. Ví dụ ta sẽ chứng minh đẳng thức:

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

Đặt

$$D_1 = A \cap (B \cup C), \quad D_2 = (A \cap B) \cup (A \cap C).$$

Lấy x tuỳ ý, cố định. Ta sẽ chỉ rõ rằng $\mu_{D_1}(x) = \mu_{D_2}(x)$. Kí hiệu

$$a = \mu_A(x), b = \mu_B(x), c = \mu_C(x).$$

Do cố định x, như vậy ứng với vectơ (a,b,c) ta chỉ cần xét 6 trường hợp sau:

	$(B \cup C)(x)$	$D_1(x)$	$(A \cap B)(x)$	$(A \cap C)(x)$	$D_2(x)$
$a \le b \le c$	c	a	a	a	a
$a \le c \le b$	b	a	a	a	a
$b \le c \le a$	c	c	b	c	c
$b \le a \le c$	c	a	b	a	a
$c \le a \le b$	b	a	a	c	a
$c \le b \le a$	b	b	b	c	b

Ví dụ ta sẽ chứng minh một mệnh đề về tính chất De Morgan

$$(A \cap B)^C = (A^C \cup B^C).$$

Cố định x, ta chỉ cần tính 2 trường hợp:

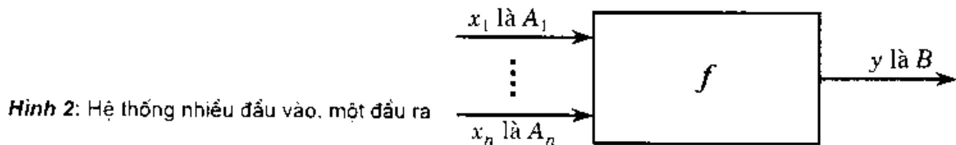
$a \le b$, khi đó $(A \cap B)^C(x) = 1 - \min(a, b) = 1 - a$, còn

$$(A^C \cup B^C)(x) = \max(1-a, 1-b) = 1-a.$$

Với $b \le a$, cũng có kết quả tương tự.

1.1.3 Nguyên lý suy rộng của Zadeh

Để trực tiếp suy rộng hàm nhiều biến, như vậy cũng sẽ cung cấp cơ sở chặt chẽ đầu tiên cho định nghĩa các hệ thống có nhiều biến vào một biến ra (Multi Input–Single Output, MISO system), nguyên lý suy rộng sau đây của Zadeh là rất quan trọng.



Hình 2: Hệ thống nhiều đầu vào, một đầu ra

Định nghĩa 1.5: Cho A_i là tập mờ với hàm thuộc μ_{A_i} trên không gian nền X_i, (i=1,2, … , n).

Khi ấy tích trực tiếp

$A = A_1 \times A_2 \times \cdots \times A_n$ là tập mờ trên không gian nền:

$X = X_1 \times X_2 \times \cdots \times X_n$ với hàm thuộc:

$$\mu_A(x) = \min\{\ \mu_{A_1}(x_1), \mu_{A_2}(x_2), \ \ldots, \mu_{A_n}(x_n)\ \},$$

trong đó $x = (x_1, x_2, \cdots, x_n)$.

Bây giờ chúng ta xét hệ thống như trên hình 2.

Nguyên lý suy rộng của Zadeh (Zadeh' extention principle): Giả sử mỗi biến vào x_i lấy giá trị là A_i (i= 1,2, $\cdots$, n) với A_i là tập mờ trên không gian nền X_i với hàm thuộc $\mu_{A_i}(x_i)$. Hàm $f: X \to Y$ chuyển các giá trị đầu vào A_i thành giá trị đầu ra B. Khi đó B sẽ là tập mờ trên Y với hàm thuộc $\mu_B(x)$ được tính theo công thức sau:

$$\mu_B(x) = \begin{cases} \sup\{\min(\mu_{A_1}(x_1), \ \ldots \ ,\mu_{A_n}(x_n) : x \in f^{-1}(y)\} & \text{nếu } f^{-1}(y) \neq \varnothing \\ 0 & \text{nếu } f^{-1}(y) = \varnothing \end{cases}$$

ở đây

$$f^{-1}(y) = \{\ x = (x_1, x_2, \cdots, x_n) \in X : f(x) = y\}$$

1.1.4 Số mờ

Chúng ta sẽ dùng các số mờ theo định nghĩa sau:

Định nghĩa 1.6: Tập mờ M trên đường thẳng số thực $\mathbb{R}^1$ là một *số mờ*, nếu

a) M chuẩn hóa, tức là có điểm x' sao cho $\mu_M(x') = 1$,

b) Ứng với mỗi $\alpha \in \mathbb{R}^1$, tập mức $\{\ x: \mu_M(x) \geq \alpha\ \}$ là đoạn đóng trên $\mathbb{R}^1$.

c) $\mu_M(x)$ là hàm liên tục.

Người ta thường dùng các số mờ dạng tam giác, hình thang và dạng hàm Gauss.

Số mờ tam giác được xác định bởi 3 tham số. Khi đó hàm thuộc của số mờ tam giác $M(a,b,c)$ cho bởi:

$$\mu_M(z) = \begin{cases} 0 & \text{nếu} & z \leq a \\ z-a/b-a & \text{nếu} & a \leq z \leq b \\ 1 & \text{nếu} & z = b \\ z-b/c-b & \text{nếu} & b \leq z \leq c \\ 0 & \text{nếu} & c \leq z \end{cases}$$

Còn số mờ hình thang $M(a,b,c,d)$ được xác định bởi 4 tham số có hàm thuộc dạng sau:

$$\mu_M(z) = \begin{cases} 0 & \text{nếu} & z \leq a \\ z-a/b-a & \text{nếu} & a \leq z \leq b \\ 1 & \text{nếu} & b \leq z \leq c \\ z-c/d-c & \text{nếu} & c \leq z \leq d \\ 0 & \text{nếu} & d \leq z \end{cases}$$

Còn hàm thuộc của số mờ dạng hàm Gauss (dạng hình chuông) cho bởi:

$$\mu_M(z) = \begin{cases} e^{-\frac{(z-a)^2}{2}} & \text{nếu } |z-a| \le d_a \\ 0 & \text{nếu } |z-a| \ge d_a \end{cases}$$

d_a là số dương được chọn thích hợp.

Định lý 1.1 (Doubois, Prade 1980): Nếu M, N là 2 số mờ thì phép cộng suy rộng $M \oplus N$ có hàm thuộc cho bởi

$$\mu_{M \oplus N}(z) = \max(\min(\mu_M(x), \mu_N(y)) : x + y = z), \text{ với mọi } z,$$

cũng là số mờ.

Khi M, N là 2 số mờ hình thang $M(a_m, b_m, c_m, d_m)$, $N(a_n, b_n, c_n, d_n)$ thì:

$$M \oplus N(a_m + a_n, b_m + b_n, c_m + c_n, d_m + d_n).$$

Định nghĩa tập mờ đối: Nếu A là tập mờ trên $\mathbb{R}^1$ với hàm thuộc $\mu_A(z)$ thì tập đối $-A$ cũng là tập mờ trên $\mathbb{R}^1$ có hàm thuộc cho bởi $\mu_{-A}(z) = \mu_A(-z)$.

Nhận xét nếu M là số mờ thì $-M$ cũng là số mờ. Hơn nữa M là số mờ hình thang thì $-M$ cũng là số mờ hình thang.

Định nghĩa phép trừ suy rộng: Cho M, N là 2 số mờ thì ta có thể định nghĩa

$$M - N = M \oplus (-N).$$

1.2 Logic mờ

Bất kỳ một người nào có ít nhiều tri thức đều hiểu rằng ngay trong những suy luận đời thường cũng như trong các suy luận khoa học chặt chẽ, logic toán học đã đóng vai trò rất quan trọng.

Nhưng đáng tiếc, chiếc áo logic toán học cổ điển đã quá chật hẹp đối với những ai mong muốn tìm kiếm những cơ sở vững chắc cho những suy luận phù hợp hơn với những bài toán nảy sinh từ nghiên cứu và thiết kế những hệ thống phức tạp, đặc biệt là những cố gắng đưa những suy luận giống như cách con người vẫn thường sử dụng vào các lĩnh vực trí tuệ nhân tạo (chẳng hạn, trong các hệ chuyên gia, các hệ hỗ trợ quyết định, các bộ phần mềm lớn, v.v...) hay vào trong công việc thiết kế và điều khiển, vận hành các hệ thống lớn, phức tạp sao cho kịp thời và hiệu quả.

Trong sự phát triển đa dạng của các hệ mờ, dựa trên cách tiếp cận mới của lý thuyết tập mờ, logic mờ giữ một vai trò rất cơ bản. Trong chương này chúng tôi sẽ hiểu logic mờ theo nghĩa đủ "hẹp" - đó là phần trực tiếp suy rộng logic mệnh đề cổ điển thông qua việc trình bày một số công cụ chủ chốt của logic mờ: các liên kết logic cơ bản.

1.2.1 Ôn nhanh về logic mệnh đề cổ điển

Ta sẽ kí hiệu $\mathcal{P}$ là tập hợp các mệnh đề và $P, P_1, Q, Q_1, \cdots$ là những mệnh đề .Với mỗi mệnh $P \in \mathcal{P}$, ta gán một trị $v(P)$ là giá trị chân lý (*truth value*) của mệnh đề. Logic cổ điển đề nghị $v(P) = 1$ nếu P là đúng (*T–true*), $v(P) = 0$ nếu P là sai (*F–false*).

Trên $\mathcal{P}$ chúng ta xác định trước tiên ba phép toán cơ bản và rất trực quan:

- Phép tuyển: P OR Q, kí hiệu $P \vee Q$, đó là mệnh đề "hoặc P hoặc Q",
- Phép hội: P AND Q, kí hiệu $P \wedge Q$, đó là mệnh đề "vừa P vừa Q",
- Phép phủ định: NOT P, kí hiệu $\neg P$, đó là mệnh đề "không P".

Dựa vào 3 phép toán logic cơ bản này người ta đã định nghĩa nhiều phép toán khác, nhưng đối với chúng ta quan trọng nhất là phép kéo theo (*implication*), kí hiệu là $P \Rightarrow Q$.

Định nghĩa 2.1: Khi sử dụng các liên kết logic: phép tuyển, phép hội, phép phủ định, phép kéo theo và phép tương đương ($\Leftrightarrow$),giá trị chân lý của mệnh đề hệ quả được xác định phụ thuộc vào giá trị chân lý của các mệnh đề gốc P, Q cho trong bảng sau:

Bảng 1

P	Q	$\neg P$	$\vee$	$\wedge$	$\Rightarrow$	$\Leftrightarrow$
1	1	0	1	1	1	1
1	0	0	1	0	0	0
0	1	1	1	0	1	0
0	0	1	0	0	1	1

Sử dụng những định nghĩa trên, trong logic cổ điển,các luật suy diễn quan trọng sau đây giữ vai trò rất quyết định trong các lập luận truyền thống. Đó là các luật:

- modus ponens: $(P \wedge (P \Rightarrow Q)) \Rightarrow Q$
- modus tollens: $((P \Rightarrow Q) \wedge \neg Q) \Rightarrow \neg P$
- syllogism: $((P \Rightarrow Q) \wedge (Q \Rightarrow R)) \Rightarrow (P \Rightarrow R)$
- contraposition: $(P \Rightarrow Q) \Rightarrow (\neg Q \Rightarrow \neg P)$

Mệnh đề 2.1: Luật modus ponen luôn đúng trong logic cổ điển

Chứng minh: Ta chỉ cần tính giá trị chân lý của $(P \wedge (P \Rightarrow Q)) \Rightarrow Q$. Thật vậy:

P	Q	$P \Rightarrow Q$	$P \wedge (P \Rightarrow Q)$	$(P \wedge (P \Rightarrow Q)) \Rightarrow Q$
1	1	1	1	1
1	0	0	0	1
0	1	1	0	1
0	0	1	0	1

Mệnh đề 2.2: Luật modus tollens $((P \Rightarrow Q) \wedge \neg Q) \Rightarrow \neg P$ luôn đúng trong logic cổ điển.

Chứng minh: Ta chỉ cần tính giá trị chân lý của $((P \Rightarrow Q) \wedge \neg Q) \Rightarrow \neg P$. Thật vậy:

P	Q	$\neg P$	$\neg Q$	$P \Rightarrow Q$	$(P \Rightarrow Q) \wedge \neg Q$	$((P \Rightarrow Q) \wedge \neg Q) \Rightarrow \neg P$
1	1	0	0	1	0	1
1	0	0	1	0	0	1
0	1	1	0	1	0	1
0	0	1	1	1	1	1

Ta hãy lấy luật modus ponens làm ví dụ. Luật này có thể giải thích như sau: Nếu mệnh đề P là đúng và nếu định lý "P kéo theo Q" đúng, thì mệnh đề Q cũng đúng.

Tương tự người đọc có thể lý giải cho các luật khác.

1.2.2 Mấy phép toán cơ bản của logic mờ

Năm 1973 L.Zadeh ([2]) đã chính thức định nghĩa và làm việc với các liên kết logic mờ cơ bản, đồng thời với việc đưa ra khái niệm *biến ngôn ngữ* đã bước đầu ứng dụng vào suy diễn mờ. Đây là bước khởi đầu rất quan trọng tính toán các suy diễn dùng logic mờ trong các hệ mờ.

Thật tự nhiên để có thể tiến hành mô hình hoá các hệ thống có nhiều thông tin bất định và nhiều tri thức còn mập mờ và tìm cách biểu diễn các quy luật vận hành trong các hệ thống này, chúng ta cần suy rộng các phép liên kết logic cơ bản (*logic connectives*) với các mệnh đề có giá trị chân lý $v(P)$ nhận trong đoạn $[0,1]$, (thay cho quy định $v(P)$ chỉ nhận giá trị 1 hoặc 0 như trước đây).

Chúng ta sẽ đưa vào các phép toán cơ bản của logic mờ qua con đường tiên đề hoá. Cho các mệnh đề $P, Q, P_1 \ldots$, giá trị chân lý $v(P)$, $v(Q)$, $v(P_1) \ldots$ sẽ nhận trong đoạn $[0,1]$.

Phần tiếp theo của phần 1.2.2 sẽ trình bày bốn phép liên kết cơ bản nhất.

1.2.2.1 Phép phủ định

Bây giờ chúng ta cho dạng toán học của phép toán này.

Định nghĩa 2.2: Hàm $n:[0,1]\rightarrow[0,1]$ không tăng thoả mãn các điều kiện $n(0)=1$, $n(1)=0$, gọi là *hàm phủ định* (negation – hay là phép phủ định).

Chúng ta có thể xét thêm vài tiên đề khác.

Định nghĩa 2.3:

a) Hàm phủ định n là *chặt* nếu nó là hàm liên tục và giảm chặt.

b) Hàm phủ định n là *mạnh* nếu nó là *chặt* và thoả mãn $n(n(x))=x$, $\forall x\in[0,1]$.

Ta hãy cho vài ví dụ và tính chất.

– Hàm phủ định thường dùng $n(x)=1-x$ (Zadeh [2]).

– Hàm $n(x)=1-x^2$

– Họ phủ định (Sugeno) $N_\lambda(x)=(1-x)/(1+\lambda x)$, với $\lambda>-1$.

Rõ ràng $(1-x)$ là phủ định mạnh. còn $(1-x^2)$ là một phủ định chặt, nhưng không mạnh. Còn với họ Sugeno ta có mệnh đề sau.

Mệnh đề 2.3: Với mỗi $\lambda > -1$, $N_\lambda(x)$ là hàm phủ định mạnh.

Chứng minh: Thật vậy, do $1+\lambda>0$ với $x_1<x_2$, $\lambda x_1+x_1< \lambda x_2+x_2$ điều này tương đương với $N_\lambda(x_1)>N_\lambda(x_2)$.

Hơn nữa $N_\lambda(N_\lambda(x))=((1+\lambda x)-(1-x)/(1+\lambda x)+\lambda(1-x))=x$, với mỗi $0\le x\le 1$.

Để thuận lợi chúng ta cần thêm hai định nghĩa sau.

Một cách định nghĩa phần bù của một tập mờ: Cho Ω là *không gian nền*, một *tập mờ* *A* trên Ω tương ứng với hàm thuộc $A: \Omega\to[0,1]$.

Định nghĩa 2.6: Cho n là hàm phủ định. phần bù A^C của tập mờ *A* là một tập mờ với hàm thuộc cho bởi $A^C(a)=n(A(a))$, với mỗi $a\in\Omega$.

Rõ ràng định nghĩa phần bù chỗ trong phần 1.1. là trường hợp riêng khi $n(x)$ là hàm phủ định thường dùng.

1.2.2.2Phép hội

Phép hội (vẫn quen gọi là phép AND – *conjunction*) là một trong mấy phép toán logic cơ bản nhất. Thông thường người ta xét các tiên đề sau:

td1. $v(P_1 \text{ AND } P_2)$ chỉ phụ thuộc vào $v(P_1)$, $v(P_2)$

td2. Nếu $v(P_1) = 1$ thì $v(P_1 \text{ AND } P_2) = v(P_2)$, với mọi mệnh đề P_2

td3. Giao hoán: $v(P_1 \text{ AND } P_2) = v(P_2 \text{ AND } P_1)$

td4. Nếu $v(P_1) \le v(P_2)$, thì $v(P_1 \text{ AND } P_3) \le v(P_2 \text{ AND } P_3)$, với mọi mệnh đề P_3

td5. Kết hợp: $v(P_1 \text{ AND } (P_2 \text{ AND } P_3)) = v((P_1 \text{ AND } P_2) \text{ AND } P_3)$

Khi diễn đạt *phép hội mờ* như một hàm số $T:[0,1]^2 \to [0,1]$, chúng ta cần tới định nghĩa sau :

Định nghĩa 2.7: Hàm $T: [0,1]^2 \to [0,1]$ là một *t–chuẩn* (*chuẩn tam giác* hay *t–norm*) nếu thoả mãn các điều kiện sau:

a) $T(1,x)=x$, với mọi $0\le x\le 1$.

b) T có tính giao hoán, tức là $T(x,y) = T(y,x)$, với mọi $0\le x,y\le 1$,

c) T không giảm theo nghĩa $T(x,y)\le T(u,v)$, với mọi $x\le u, y\le v$,

d) T có tính kết hợp: $T(x,T(y,z)) = T(T(x,y),z)$ với mọi $0\le x,y,z\le 1$.

Từ những tiên đề trên chúng ta suy ra ngay $T(0,x)$. Hơn nữa tiên đề d) đảm bảo tính thác triển duy nhất cho hàm nhiều biến.

Ví dụ 2.1:

1) Min (Zadeh): $T_1(x,y) = \min(x,y)$

2) $T_2(x,y) = \dfrac{xy}{x+y-xy}$

3) t–chuẩn dạng tích: $T_3(x,y) = xy$,

4) $T_4(x,y) = \dfrac{xy}{2-(x+y-xy)}$

5) t–chuẩn Lukasiewicz: $T_L(x,y) = \max\{\, x+y-1\ ,\ 0 \,\}$,

6) t–chuẩn yếu nhất (drastic product): $Z(x,y) = \begin{cases} \min(x,y) & \text{nếu} \quad \max(x,y)=1 \\ 0 \quad \text{nếu} & \max(x,y)<1 \end{cases}$

Bây giờ chúng ta xét vài tính chất của t–chuẩn:

Mệnh đề 2.4: Với mỗi t-chuẩn T thì:

$$T_5(x,y) \le T(x,y) \le T_1(x,y) = \min(x,y) \quad \text{với mọi } 0 \le x,y \le 1$$

Chứng minh: Thật vậy.

Nếu $\max(x,y)=1$.

Khi $x=1$, $T(1,y)=y=\min(x,y)$ hay $T_5(x,y)=T(x,y)=T_1(x,y)$. Tương tự nếu $y=1$.

Nếu $\max(x,y)<1$, $T_5(x,y)=0<T(x,y)$.

Giả sử $y=\min(x,y)$, khi đó $T(x,y) \le T(1,y)=y=T_1(x,y)$. Tương tự nếu $x=\min(x,y)$.

Định nghĩa 2.8:

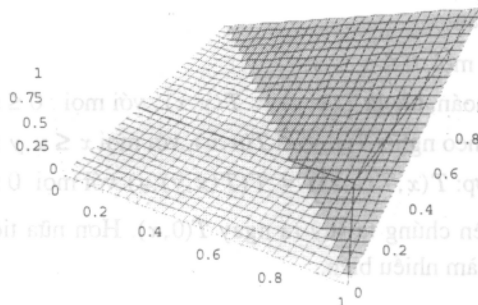
a) Một t-chuẩn T gọi là liên tục nếu T là hàm liên tục trên $[0,1]^2$

b) Hàm T gọi là Archimed nếu $T(x,x)<x$ với mọi $0<x<1$

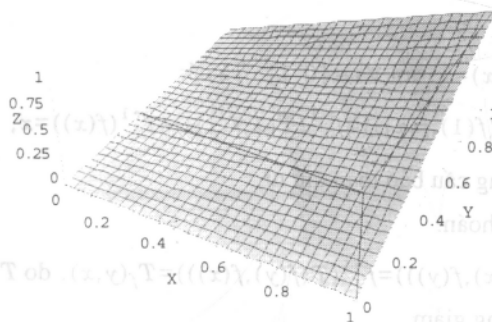
c) Hàm T gọi là chặt nếu T tăng chặt trên $(0,1)^2$

Sau đây là đồ thị của một số hàm t-chuẩn:

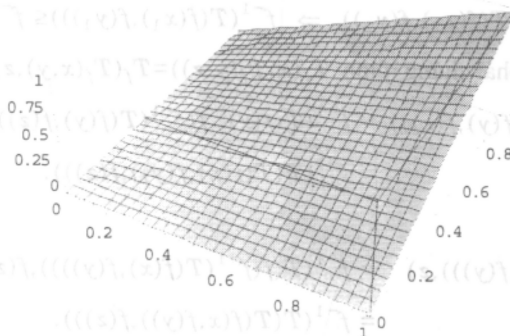
$T_L(x,y) = \max(0, x+y-1)$.



$$T_4(x,y) = \frac{xy}{2-(x+y-xy)}$$



$$T_3(x,y) = xy$$



Không khó khăn kiểm tra các t-chuẩn $T_1(x,y)$, $T_2(x,y)$, $T_3(x,y)$, $T_4(x,y)$, t-chuẩn Lukasiewicz là liên tục và $T_2(x,y)$, $T_3(x,y)$, $T_4(x,y)$ là Archimed. Thật vậy:

1) $T_4(x,y) = \dfrac{xy}{2-(x+y-xy)}$ là Archimed vì $T_2(a,a) = \dfrac{a^2}{2-(2-(2a-a^2))}$ và do:

$a^2-2a+2 = (a-1)^2+1 > 1 \Rightarrow \dfrac{a2}{2-(2a-a2)} < a^2 \le a$

Vậy $T_2(a,a) < a$ với mọi $a \in (0,1)$.

2) $T_3(x,y) = xy$ là chặt vì $0 \le x_1 < x_2$, $0 \le y_1 < y_2$, ta có $x_1y_1 < x_2y_2$.

3) $T_5(x,y) = \min(x,y)$ là một hàm liên tục trên $[0,1]^2$, nên t-chuẩn T là liên tục. Hơn thế nữa, ta luôn có $T_5(x,x) = \min(x,x) = x$.

Hàm sinh của lớp toán tử t-chuẩn: Cho f là một đẳng cấu bảo toàn thứ tự từ $[0,1] \to [0,1]$, ta có định lí sau:

Định lý 2.1: Cho T là một t-chuẩn, ta xác định $T_f:[0,1]\times[0,1]\to[0,1]$ bằng định nghĩa:

$T_f(x,y) = f^{-1}(T(f(x), f(y)))$, với mọi $0\le x,y \le 1$.

Khi đó T_f là một t-chuẩn. Nếu T là Archimed thì T_f là Archimed.

Chứng minh:

1) Chứng minh $T_f(1,x)=x$, với $\forall x\in[0,1]$. Ta có:

$$T_f(1,x)=f^{-1}(T(f(1),f(x)))=f^{-1}(T(1,f(x)))=f^{-1}(f(x))=x,$$

với $\forall x\in[0,1]$, vì f đẳng cấu bảo toàn thứ tự.

2) Kiểm tra tính giao hoán.

$T_f(x,y)=f^{-1}(T(f(x),f(y)))=f^{-1}(T(f(y),f(x)))=T_f(y,x)$, do T có tính giao hoán.

3) Tính đơn điệu không giảm

Với $x_1\le x_2$ và $y_1\le y_2$ do f là đẳng cấu bảo toàn thứ tự nên f^{-1} cũng là đẳng cấu bảo toàn thứ tự và $f(x_1)\le f(x_2)$, $f(y_1)\le f(y_2)$. Lại do T là t-chuẩn nên:

$$T(f(x_1,f(y_1))\le T(f(x_2),f(y_2)) \;\Rightarrow\; f^{-1}(T(f(x_1),f(y_1)))\le f^{-1}(T(f(x_2),f(y_2))).$$

4) Tính kết hợp: Ta phải chứng minh $T_f(x,T_f(y,z))=T_f(T_f(x,y),z)$. Ta có vế trái:

$$\begin{aligned}T_f(x,f^{-1}(T(f(y),f(z))) &= f^{-1}(T(f(x),\ f(f^{-1}(T(f(y),f(z)))))\\ &= f^{-1}(T(T(f(x),f(y)),f(z))).\end{aligned}$$

Vế phải là:

$$\begin{aligned}T_f(f^{-1}(T(f(x),f(y))),z) &= f^{-1}(T(f(f^{-1}(T(f(x),f(y)))),f(z)))\\ &= f^{-1}(T(T(f(x,f(y)),f(z))).\end{aligned}$$

Ta thấy vế trái bằng vế phải.

5) Kiểm tra T là Archimed thì T_f là Archimed. Theo giả thiết:

T là Archimed $\Rightarrow T(a,a)<a$ với mọi $a\in[0,1]$

Ta có:

$$T_f(a,a) = f^{-1}(T(f(a),f(a))) = f^{-1}(c)<f^{-1}(f(a)) = a,$$

vì $\quad T(f(a),f(a)) = c < f(a)$

Ví dụ 2.2: Xét $T(x,y)=xy$. Khi đó ta có nhiều cách tạo ra T_f

Cách 1: Chọn $f(x)=x$ ta có $f^{-1}(x)=x$; $T(x,y)=xy=f(x)f(y)=f^{-1}(f(x)f(y))$. Vậy:

$$T_f(x,y) = f^{-1}(f(x)f(y)) = xy$$

Cách 2: Chọn $f(x)=x^n \Rightarrow f^{-1}(x)=\sqrt[n]{x}$, khi đó:

$$T_f(x,y) = f^{-1}(f(x)f(y)) = f^{-1}(x^n,y^n) = xy$$

Ví dụ 2.3: Xét $T(x,y)= \dfrac{xy}{a+(1-a)(x+y-xy)}$. Ta chọn $f(x)= \dfrac{x}{x+a(1-x)}$

Với $a>0$ ta thấy $f(x)$ liên tục và $f(0)=0/a=0$; $f(1)=1$. Vậy $f(x)$ là đẳng cấu bảo toàn thứ tự. Ta có $f^{-1}(x)= \dfrac{ax}{ax-x+1}$. Theo định lý trên ta có:

$$T_f(x,y) = f^{-1}(T(f(x),f(y))) = f^{-1}(f(x)\cdot f(y)).$$

Nếu $x=1/2$ thì $f^{-1}(1/2)= f(\frac{1}{2}) = \dfrac{a(\frac{1}{2})}{a(\frac{1}{2})-\frac{1}{2}+1} = \dfrac{1}{1+\frac{1}{a}}$. Ta thấy $T(x,y)$ phụ thuộc vào $a>0$.

Xét một số trường hợp đặc biệt:

a) $a=1 \Rightarrow f(x)=x \Rightarrow T_f(x,y)=xy$

b) $a=1/2 \Rightarrow f(x)= \dfrac{2x}{1+x} \Rightarrow f^{-1}(x)= \dfrac{x}{2-x}$. Vậy $T_f(x,y)= \dfrac{2xy}{1+(x+y-xy)}$.

c) $a=2 \Rightarrow f(x)= \dfrac{x}{2-x} \Rightarrow T_f(x,y)= \dfrac{xy}{2-(x+y-xy)} = T_2(x,y)$

Định nghĩa tổng quát phép giao của 2 tập mờ: Cho hai tập mờ A, B trên cùng không gian nền X với hàm thuộc tương ứng là $A(x)$, $B(x)$. Cho T là một t-chuẩn.

Định nghĩa 2.9: Ứng với t-chuẩn T, tập giao của hai tập mờ A, B là một tập mờ $(A\cap_T B)$ trên X với hàm thuộc cho bởi:

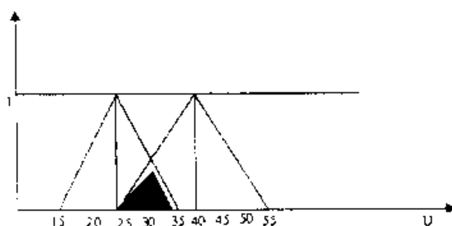
$$(A\cap_T B)(x) = T(A(x),B(x)), \quad \forall x\in X$$

Việc lựa chọn phép giao tương ứng với t-chuẩn T nào tuỳ thuộc vào bài toán được quan tâm.

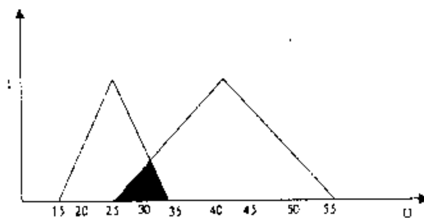
Ví dụ 2.4: Cho U là không gian nền: $U=[0,120]$ - thời gian sống.

A = {Những người ở tuổi trung niên} ; B = {Những người ở tuổi thanh niên}

Khi đó giao của hai tập mờ A và B, khi sử dụng $T(x,y)=\min(x,y)$ và $T(x,y)=xy$ chúng sẽ được biểu diễn trên hình vẽ như sau:



Dạng tích $T(x,y)=xy$



Dạng $T(x,y)=\min(x,y)$

1.2.2.3 Phép tuyển

Giống như phép hội, phép tuyển hay toán tử logic OR (*disjunction*) thông thường cần thoả mãn các tiên đề sau:

Định nghĩa 2.10: Hàm $S:[0,1]^2 \to [0,1]$ gọi là *phép tuyển* (OR suy rộng) hay là *t–đối chuẩn* (*t–conorm*) nếu thoả mãn các tiên đề sau:

a) $S(0,x) = x$ với mọi $x \in [0,1]$

b) S có tính giao hoán: $S(x,y) = S(y,x)$ với mọi $0 \le x,y) \le 1$,

c) S không giảm: $S(x,y) \le S(u,v)$ với mọi $0 \le x \le u \le 1$ và $0 \le y \le v \le 1$

d) S có tính kết hợp: $S(x,S(y,z)) = S(S(x,y),z)$ với mọi $0 \le x,y,z \le 1$

Từ định nghĩa ta thấy: $S(0,1) \le S(x,1) \Leftrightarrow 1 \le S(x,1) \le 1 \Rightarrow S(x,1) = 1$.

Ví dụ 2.5:

$$S_0(x,y) = \max(x,y)$$

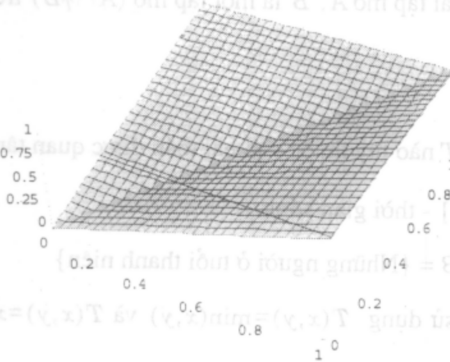
$$S_1(x,y) = x+y-xy$$

$$S_2(x,y) = \min(1,x+y)$$

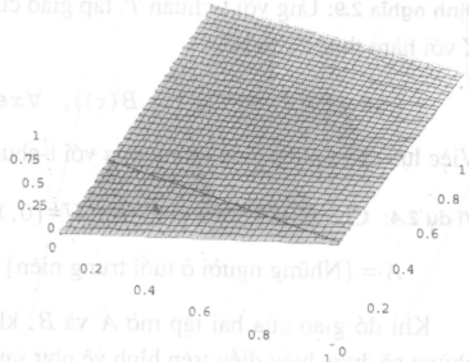
$$S_3(x,y) = \max_1(x,y) = \begin{cases} \max(x,y) & \text{khi } x+y=1 \\ 1 \text{ khi } x+y \ne 1 \end{cases}$$

$$S_4(x,y) = \begin{cases} \max(x,y) & \text{khi } \min(x,y)=0 \\ 1 \text{ khi } \min(x,y) \ne 0 \end{cases}$$

Các hàm này đều là các t-đối chuẩn. Sau đây là đồ thị của một số hàm t-đối chuẩn:

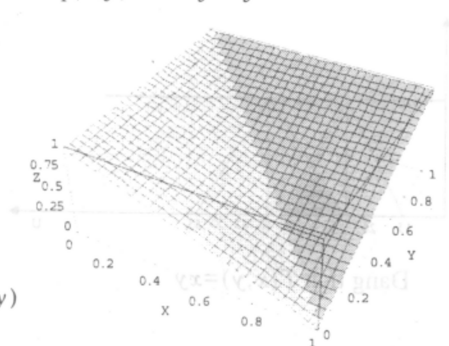


$S_0(x,y) = \max(x,y)$



$S_1(x,y) = x+y-xy$





$S_2(x,y) = \min(1,x+y)$

Tiếp tục ta sẽ xem xét một số tính chất của t-đối chuẩn

Định lý 2.2: Với S là một t-đối chuẩn bất kỳ thì bất đẳng thức sau luôn đúng với mọi $x, y \in [0,1]$:

a) $S_0(x,y) \leq S(x,y) \leq S_4(x,y)$.

b) $S_0 \leq S_1 \leq S_2 \leq S_4$

Chứng minh:

a) Khi $x=1$, ta có $S(1,y)=1$, $S_0(1,y)=\max(1,y)=1=S_4(x,y)$. Khi $y=1$, tương tự $S_0(x,1)=S(x,1)=S_4(x,y)$. Khi $x=0$, tương tự ta cũng có $S_0(0,y)=S(0,y)=S_4(0,y)$. Khi $y=0$, ta cũng có $S_0(x,0)=S(x,0)=S_4(x,0)$.

- Xét $0<x,y<1$: Ta có $\min(x,y)\neq 0$, suy ra $S_4(x,y)=1$

  Nếu $x>y$, suy ra $S_0((x,y))=x < S_4(x,y)=1$. Thấy $x=S(0,x)\leq S(y,x)$ (tính không giảm). Vì vậy $S_0(x,y)<S(x,y)<S_4(x,y)$.

  Nếu $y>x$: $S_0((x,y))=y < S_4(x,y)=1$. Ta thấy $y=S(0,y)<S(x,y)$. Do đó cũng có $S_0(x,y)<S(x,y)<S_4(x,y)$.

Vậy $\forall x,y\in [0,1]$ thì $S_0(x,y)<S(x,y)<S_4(x,y)$.

b) Phần a) ta đã chứng minh được rằng: $S_0(x,y)<S(x,y)<S_4(x,y)$. Giờ đây ta phải chứng minh: $S_0\leq S_1\leq S_2\leq S_4$ và $S_0\leq S_2\leq S_3\leq S_4$

- Chứng minh $S_0\leq S_1$: Xét $x\geq y$ suy ra

  $S_0(x,y)=\max(x,y)=x$, và $S_1(x,y)=x+y-xy=x+(1-x)y$

  Thấy $0\leq x,y \leq 1 \Rightarrow 1-x\geq 0 \Rightarrow (1-x)y\geq 0 \Rightarrow x+(1-x)y\geq x$

  Vậy $S_0(x,y)\leq S_1(x,y)$

  Tương tự, với $y\geq x$ $S_0(x,y)\leq S_1(x,y)$. Vậy $\forall x,y\in [0,1]$ luôn có $S_0(x,y)\leq S_1(x,y)$.

- Chứng minh $S_1\leq S_2$: Xét $x+y\leq 1$. Khi đó $S_2(x,y)=x+y$. Thấy $0\leq x,y\leq 1$, suy ra

  $1-x\leq 1 \Rightarrow (1-x)y\leq y \Rightarrow x+(1-x)y \leq x+y$

  Vậy $S_1(x,y)\leq S_2(x,y)$.

  Xét $x+y>1$, khi đó $S_2(x,y)=1$. Do

  $x,y\in [0,1] \Rightarrow (1-x)(1-y)\geq 0 \Leftrightarrow x+y-1\leq xy \Leftrightarrow x+y-xy\leq 1$.

  Vậy $S_1(x,y)\leq S_2(x,y)$. Do đó $\forall x,y\in [0,1]$ luôn có $S_1(x,y)\leq S_2(x,y)$.

- Chứng minh $S_2\leq S_4$: Xét $x=0 \Rightarrow x+y\leq 1$, do đó $S_3(x,y)=\max(x,y)$, lại do $x=0$, suy ra

  $\min(x,y)=0 \Rightarrow S_4(x,y)=\max(x,y)=S_3(x,y)$.

Tương tự xét $y=0$, ta cũng lại có kết quả như trên. Xét tiếp với $x\neq 0$, $y\neq 0$. Khi đó: $S_4(x,y)=1$.

Khi $x+y<1 \Rightarrow S_3(x,y)=\max(x,y)\leq 1=S_4(x,y)$

Khi $x+y>1 \Rightarrow S_3(x,y)=1=S_4(x,y)$

Vậy $\forall x,y\in[0,1]$ luôn có $S_2(x,y)\leq S_4(x,y)$.

Từ các kết quả trên ta được: $S_0(x,y)\leq S_1(x,y)\leq S_2(x,y)\leq S_4(x,y)$

Từ đó ta có mệnh đề sau:

Mệnh đề 2.5: Nếu S là t-đối chuẩn thì:

$$\max(x,y) \leq S(x,y) \leq Z'(x,y) \quad \text{với mọi } 0\leq x,y\leq 1.$$

Định nghĩa 2.11: Cho S là t-đối chuẩn. Khi ấy:

a) S gọi là liên tục nếu S là hàm liên tục trên $[0,1]^2$.

b) Hàm S gọi là Archimed nếu $S(x,x)>x$ với mọi $0<x<1$.

c) S gọi là chặt nếu S tăng chặt trên $(0,1)^2$.

Ví dụ 2.6: $S_1(x,y)=x+y-xy$, là chặt vì: Giả sử $x_1<x_2$, ta có

$$S_1(x_1,y)=x_1+y-x_1y<x_2+y-x_2y=S_1(x_2,y),\ \forall y\in(0,1).$$

Mặt khác do S có tính chất giao hoán nên ta có $S_1(x_1,y_1)<S_1(x_2,y_2)$, $\forall\ 0<x_1<x_2<1$, và $\forall\ 0<y_1<y_2<1$.

$S_0(x,y)=\max(x,y)$ là một hàm liên tục trên $[0,1]^2$, nên t-đối chuẩn S là liên tục. Hơn thế nữa, ta luôn có $S_0(x,y)=\max(x,y)=x$.

$S_2(x,y)=\min\{1,\ x+y\}$ là Archimed vì $S_2(x,x)\min\{1,\ x+x\}=\min\{1,2x\}>x$.

<u>*Hàm sinh của lớp toán tử t-đối chuẩn*</u>: Cho f là một đẳng cấu bảo toàn thứ tự từ $[0,1]\rightarrow[0,1]$, ta có định lí sau:

Định lý 2.3: Cho S là một t-đối chuẩn, ta xác định:

$$S_f : [0,1]\times[0,1] \rightarrow [0,1]$$

$$S_f(x,y) = f^{-1}(S(f(x),\ f(y))).$$

Khi đó S_f là một t-đối chuẩn. Nếu S là Archimed thì S_f là Archimed.

Chứng minh: Dành cho bạn đọc như một bài tập.

Ví dụ 2.7: Xét $S(x,y)=x+y-xy$. Khi đó ta có nhiều cách tạo ra S_f, ta có thể chọn:

$f(x)=x$, khi đó

$f^{-1}(x)=x, \quad S(x,y)=x+y-xy=f(x)+f(y)-f(x)f(y)$

$\Rightarrow \quad S_f(x,y)= f^{-1}(f(x)+f(y)-f(x)f(y)) = x+y-xy$

Vậy $\quad S_f(x,y)= f^{-1}(f(x)+f(y)-f(x)f(y))$.

Định nghĩa tổng quát phép hợp của 2 tập mờ.

Định nghĩa 2.12: Cho A và B là 2 tập mờ trên không gian nền X, với hàm thuộc $A(x)$, $B(x)$ tương ứng. Cho S là một t-đối chuẩn. Phép hợp $(A \cup_S B)$ là một tập mờ trên X với hàm thuộc cho bởi biểu thức:

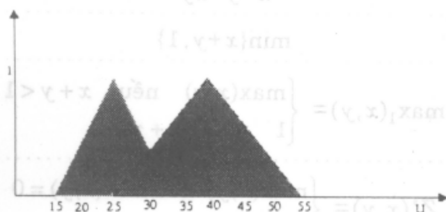
$$(A \cup_S B)(x) = S(A(x), B(x)), \quad \forall x \in X.$$

Việc lựa chọn phép hợp, tương ứng với t-đối chuẩn S nào tuỳ thuộc vào bài toán ta quan tâm.

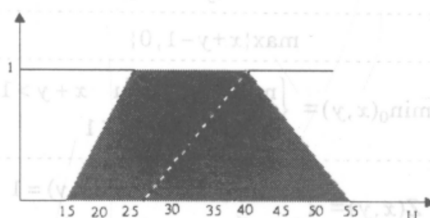
Ví dụ 2.8: Cho U là không gian nền: $U = [0,120]$ là thời gian sống.

A ={Những người ở tuổi trung niên}; B ={Những người ở tuổi thanh niên}.

Khi đó hợp của hai tập mờ A, B với $T(x,y)=\max(x,y)$ và $T(x,y)=\max(1,x+y)$. Chúng biểu diễn trên hình vẽ như sau:



Dạng max



Dạng Lukasewicz

Bộ ba De Morgan:

Trong lý thuyết tập hợp luật De Morgan nổi tiếng sau đây đã được sử dụng nhiều nơi: Cho A, B là hai tập con của X, khi đó

$$(A \cup B)^C = A^C \cap B^C$$

và $$(A \cap B)^C = A^C \cup B^C$$

Có nhiều dạng suy rộng hai đẳng thức này. Sau đây một dạng suy rộng cho logic mờ.

Định nghĩa 2.13: Cho T là t–chuẩn, S là t–đối chuẩn, n là phép phủ định mạnh. Chúng ta nói bộ ba (T, S, n) là một *bộ ba De Morgan* nếu thỏa mãn một trong 2 đẳng thức sau:

$S(x,y) = n(T(n(x),n(y)))$ hay

$T(x,y) = n(S(n(x),n(y)))$

Khi ấy ta nói T và S đối ngẫu với nhau. Quan hệ đối ngẫu giữa t-chuẩn và t-đối chuẩn có thể thấy qua định lý sau.

Định lý 2.4: Cho n là phép phủ định mạnh.

a) $S(x,y)$ là một t–đối chuẩn và $T(x,y)$ cho bởi:

$T(x,y)=nT(S(n(x),n(y))$ với mọi $0 \le x,y \le 1$.

Khi đó $T(x,y)$ là một t–chuẩn.

b) Đối ngẫu, cho $T(x,y)$ là t-chuẩn và $S(x,y)$ cho bởi

$S(x,y)=nT(T(n(x),n(y))$ với mọi $0 \le x,y \le 1$.

Khi đó $S(x,y)$ là một t–chuẩn.

Chứng minh: Dành cho bạn đọc như một bài tập.

Sau đây là mấy cặp đối ngẫu cụ thể.

Ví dụ: Chọn $n(x)=1-x$, chúng ta có các cặp sau:

$T(x,y)$	$S(x,y)$
$\min(x,y)$	$\max(x,y)$
xy	$x+y-xy$
$\max\{x+y-1,0\}$	$\min\{x+y,1\}$
$\min_0(x,y)=\begin{cases}\min(x,y) & \text{nếu } x+y>1\\ 0 & \text{nếu } x+y\le 1\end{cases}$	$\max_1(x,y)=\begin{cases}\max(x,y) & \text{nếu } x+y<1\\ 1 & \text{nếu } x+y\ge 1\end{cases}$
$Z(x,y)=\begin{cases}\min(x,y) & \text{nếu } \max(x,y)=1\\ 0 & \text{nếu } \max(x,y)\ne 1\end{cases}$	$Z'(x,y)=\begin{cases}\max(x,y) & \text{nếu } \min(x,y)=0\\ 1 & \text{nếu } \min(x,y)\ne 0\end{cases}$

1.2.2.4 Phép kéo theo

Phép kéo theo là công đoạn chủ chốt nhất của quá trình suy diễn trong mọi lập luận xấp xỉ, bao gồm cả lập luận mờ. Phép kéo theo (Implication) được xét như một mối quan hệ, một toán tử logic. Khi mô hình hoá có thể xét tới các tiên đề sau cho hàm $v(P_1 \Rightarrow P_2)$:

tđ$_0$: $v(P_1 \Rightarrow P_2)$ chỉ phụ thuộc vào giá trị $v(P_1)$, $v(P_2)$

tđ$_1$: Nếu $v(P_1) \le v(P_2)$ thì $v(P_1 \Rightarrow P_2) \ge v(P_3 \Rightarrow P_2)$ với mọi mệnh đề P_2

tđ$_2$: Nếu $v(P_1) \le v(P_3)$ thì $v(P_1 \Rightarrow P_2) \le v(P_1 \Rightarrow P_3)$ với mọi mệnh đề P_1

tđ$_3$: Nếu $v(P_1)=0$ thì $v(P_1 \Rightarrow P)=1$ với mỗi mệnh đề P

tđ$_4$: Nếu $v(P_1)=1$ thì $v(P \Rightarrow P_1)=1$ với mỗi mệnh đề P

tđ$_5$: Nếu $v(P_1)=1$ và $v(P_2)=0$ thì $v(P_1 \Rightarrow P_2)=0$

Tính hợp lí của các tiên đề này chủ yếu dựa vào logic cổ điển và những tư duy trực tiếp về phép suy diễn. Từ tiên đề I_0 ta khẳng định sự tồn tại của hàm $I(x,y)$ xác định trên $[0,1]^2$, với giá trị chân lí qua biểu thức sau:

$$v(P_1 \Rightarrow P_2) = I(v(P_1),v(P_2))$$

Định nghĩa 2.14: *Phép kéo theo (implication)* là một hàm số $I:[0,1]^2 \rightarrow [0,1]$ thỏa mãn các điều kiện sau:

a) Nếu $x \le z$ thì $I(x,y) \ge I(z,y)$ với mọi $y \in [0,1]$
b) Nếu $y \le u$ thì $I(x,y) \le I(x,u)$ với mọi $x \in [0,1]$
c) $I(0,x) = 1$ với mọi $x \in [0,1]$
d) $I(x,1) = 1$ với mọi $x \in [0,1]$
e) $I(1,0) = 0$.

<u>*Một số dạng hàm kéo theo cụ thể*</u>:

Định nghĩa 2.15: *Dạng kéo theo thứ nhất*. Cho $S(x,y)$ là một t-đối chuẩn, $n(x)$ là một phủ định mạnh . Hàm $I_S(x,y)$ xác định trên $[0,1]^2$ bằng biểu thức:

$$I_S(x,y) = S(n(x),y),\ \forall 0 \le x,y \le 1.$$

Rõ ràng ẩn ý sau định nghĩa này là công thức từ logic cổ điển $P \Rightarrow Q \Leftrightarrow (\neg P \vee Q)$

Định lý 2.5: Với bất kỳ t–chuẩn T, t–đối chuẩn S và phép phủ định mạnh n nào, I_S được định nghĩa như trên là một phép kéo theo.

<u>*Chứng minh.*</u> (Ta kiểm chứng I_S theo từng tiên đề của định nghĩa 2.14.).

a) Tiên đề I_1: Cho $x \le z$. Vì $I_S(x,y)=S(n(x),y)$. Ta chỉ xét trường hợp $x<z$, khi ấy $n(x)>n(z)$. Do t-đối chuẩn không giảm theo hai biến

$$I_S(x,y) = S(n(x),y) \ge S(n(z),y) = I_S(z,y)$$

b) Tiên đề I_2: Cho $y \le t$, khi đó $I_S(x,y)=S(nx,y) \le S(nx,t)=I_S(x,t)$, $\forall x$.

c) Tiên đề I_3: $I_S(0,x)=S(n(0),x)=I_S(x,y)=S(1,x) \ge \max(1,x)=1$, vậy $I_S(0,x)=1$, $\forall x$.

d) Tiên đề I_4: $I_S(x,1)=S(nx,1) \ge \max(nx,1)=1$, vậy $I_S(x,1)=1$, $\forall x$.

e) Tiên đề I_5: $I_S(1,0)=S(n(1),0)=S(0,0)=0$, vậy $I_S(1,0)=0$, $\forall x$.

I_S là một phép kéo theo của logic mờ thoả mãn định nghĩa 2.14.

Định nghĩa 2.16: *Dạng kéo theo thứ hai*. Cho T là một t-chuẩn, hàm $I_T(x,y)$ xác định trên $[0,1]^2$ bằng biểu thức:

$$I_T(x,y) = \sup\{ u : T(x,u) \le y,\ \forall 0 \le x,y \le 1 \}$$

Định lý 2.6: Với bất kỳ t-chuẩn T nào, I_T được định nghĩa như trên là một phép kéo theo.

Chứng minh: Ta kiểm chứng I_T theo từng tiên đề của định nghĩa 2.14.

a) Tiên đề I_1: Cho $x \le z$. Vì $I_T(x,y) = \sup\{u : T(x,u) \le y\}$. Do t-chuẩn T không giảm theo hai biến nên $T(x,u) \le T(z,u)$ và do vậy:

$$\{u : T(z,u) \le y\} \subseteq \{u : T(x,u) \le y\}$$

$$\sup\{u : T(z,u) \le y\} \le \sup\{u : T(x,u) \le y\}.$$

Hay $I_T(z,y) \le I_T(x,y)$ với mọi y. Đó chính là điều kiện I_1.

b) Tiên đề I_2: Cho $y \le t$, khi đó với mỗi cặp (x,u) ta có $T(x,u) \le y \le t$:

$$\{u : T(x,u) \le y\} \subseteq \{u : T(x,u) \le t\}$$

$$\sup\{u : T(x,u) \le y\} \le \sup\{u : T(x,u) \le t\}.$$

Hay $I_T(x,y) \le I_T(x,t)$ với mọi x. Đó chính là điều kiện I_2

c) Tiên đề I_3: $T(0,x) = x$ với bất kì u nào ta có $0 \le u \le 1$. Do vậy $T(0,u) \le x$, suy ra

$$\sup\{u : T(0,u) \le x\} = 1$$

với mọi x hay $I_T(0,x) = 1$. Đó chính là điều kiện I_3

d) Tiên đề I_4: $I_T(x,1) = 1$ là hiển nhiên với mọi x.

e) Tiên đề I_5: Do $I_T(1,0) = \sup\{u : T(1,u) \le 0\}$, điều này dẫn tới $T(1,u) = 0$. Sử dụng tính chất $T(1,u) = u$ của t-chuẩn, chỉ có $u = 0$ thoả mãn đẳng thức, tức là $T(1,0) = 0$. Vậy I_T là một phép kéo theo của logic mờ thoả mãn định nghĩa 2.14

Như đã nhận xét từ đầu, có rất nhiều con đường muốn xác định phép kéo theo. Phép kéo theo sau đây, nói chung không thoả mãn tiên đề 1, nhưng được nhiều tác giả sử dụng, ý chính của phép kéo theo này bắt nguồn từ biểu diễn phép $P \Rightarrow Q$ theo lý thuyết tập hợp.

Nếu P, Q là các mệnh đề trong logic cổ điển hay ta biểu diễn dưới dạng tập hợp trong cùng một không gian nền thì $(P \Rightarrow Q) = (\neg P \vee (P \wedge Q))$.

Sử dụng T là t-chuẩn, S là t-đối chuẩn, n là phép phủ định, thì có thể nghĩ tới dạng:

$$I(x,y) = S(T(x,y),\ n(x))$$

Lập luận tương tự khi cho P và Q trên các không gian nền khác nhau cũng có thể dẫn tới cùng dạng hàm $I(x,y)$ này.

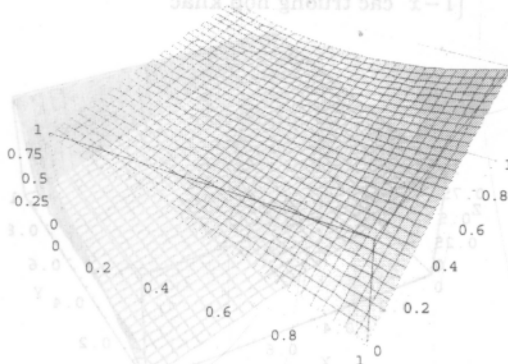
Định nghĩa 2.17: *Dạng kéo theo thứ ba*. Cho (T,S,n) là bộ ba De Morgan với n là phép phủ định mạnh, phép kéo theo thứ ba $I_S(x,y)$ xác định trên $[0,1]^2$ bằng biểu thức:

$$I_S(x,y) = S(T(x,y),n(x)),\ \forall 0 \le x,y \le 1$$

Ví dụ 2.9: Chọn $T(x,y) = xy$, $S(x,y) = x + y - xy$, ta được:

$$I_S(x,y)=S(xy,(1-x))=xy+(1-x)-xy(1-x).$$

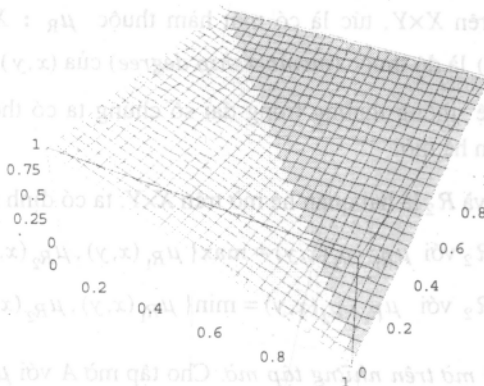
Từ đó $I_S(x,y)=1-x+x^2y$



Chọn $n(x)=1-n$, $T(x,y)=\max\{x+y-1,\ 0\}$; $S(x,y)=\min\{x+y,1\}$, có:

$$\begin{aligned} I_S(x,y) &= S\{\max(x+y-1,0),\ 1-x\} \\ &= \min\{\max(x+y-1,0)+(1-x),1\} \\ &= \min\{\max(y,1-x),1\}. \end{aligned}$$

Do $y\le 1$. nên luôn có: $\max(y,1-x)\le 1$. Khi đó ta được: $I_S(x,y)=\max(1-x,y)$



Chọn $T(x,y)=Z(x,y)$, $S(x,y)=Z'(x,y)$. *Xét lần lượt các trường hợp:*

- Nếu $\max(x,y)\neq 1$ thì ta có $T(x,y)=Z(x,y)=0$, khi đó:

  $$\begin{aligned} I_S(x,y) &= S(T(x,y),\ n(x)) = Z'(0,1-x) \\ &= \max(0,1-x) = 1-x \qquad (\text{do } \min(0,1-x)=0) \end{aligned}$$

- Nếu $x=1$, ta có $T(x,y)=Z(1,y)=\min(1,y)=y$, khi đó

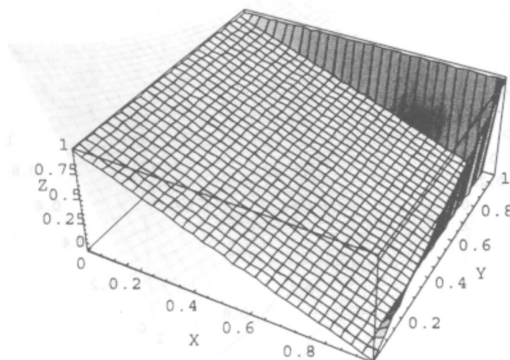
  $$I_S(x,y) = S(T(x,y),\ n(x)) = Z'(y,1-1) = Z'(y,0) = y$$

- Nếu $y=1$; $x<1$ ta có $T(x,y)=Z(x,1)=\min(x,1)=x$. khi đó

  $$I_S(x,y) = S(T(x,y),\ n(x)) = Z'(1,\ 1-x) = Z'(y,0) = y$$

  $= 1$ vì ($x < 1$, nên $\min(1,1-x)=1-x>0$).

Tóm lại ta có: $I_S(x,y) = \begin{cases} y & \text{khi } x=1 \\ 1 & \text{khi } y=1 \\ 1-x & \text{các trường hợp khác} \end{cases}$



1.3 Quan hệ mờ

1.3.1 Một số khái niệm của quan hệ mờ

Định nghĩa 3.1: Cho X, Y là hai không gian nền. R gọi là một *quan hệ mờ* trên $X \times Y$ nếu R là một tập mờ trên $X \times Y$, tức là có một hàm thuộc $\mu_R : X \times Y \to [0,1]$, ở đây $\mu_R(x,y) = R(x,y)$ là độ thuộc (*membership degree*) của (x,y) vào quan hệ R.

Như những quan hệ thông thường trong đại số chúng ta có thể xét những khái niệm quen thuộc cho các quan hệ mờ.

Định nghĩa 3.2: Cho R_1 và R_2 là hai quan hệ mờ trên $X \times Y$, ta có định nghĩa

a) Quan hệ $R_1 \cup R_2$ với $\mu_{R_1 \cup R_2}(x,y) = \max\{\mu_{R_1}(x,y), \mu_{R_2}(x,y)\}$, $\forall (x,y) \in X \times Y$.

b) Quan hệ $R_1 \cap R_2$ với $\mu_{R_1 \cap R_2}(x,y) = \min\{\mu_{R_1}(x,y), \mu_{R_2}(x,y)\}$, $\forall (x,y) \in X \times Y$.

Định nghĩa 3.3: *Quan hệ mờ trên những tập mờ.* Cho tập mờ A với $\mu_A(x)$ trên X, tập mờ B với $\mu_B(x)$ trên Y. Quan hệ mờ trên các tập mờ A và B là quan hệ mờ R trên $X \times Y$ thoả mãn điều kiện:

$\mu_R(x,y) \le \mu_A(x)$, $\forall y \in Y$.

$\mu_R(x,y) \le \mu_B(x)$, $\forall x \in X$.

Định nghĩa 3.4: Cho quan hệ mờ R trên $X \times Y$.

- *Phép chiếu* của R lên X là: $\text{proj}_X R = \{(x, \max_y \mu_R(x,y) : x \in X\}$
- *Phép chiếu* của R lên Y là: $\text{proj}_Y R = \{(y, \max_x \mu_R(x,y) : y \in X\}$
- *Thác triển* R lên không gian tích $X \times Y \times Z$ là:

$\text{ext}_{XYZ}\, R = \{(x,y,z), \mu_{\text{ext}}(x,y,z) = \mu_R(x,y), \forall z \in Z\}$.

Định nghĩa 3.5 (*phép hợp thành*): Cho R_1 là quan hệ mờ trên $X \times Y$ và R_2 là quan hệ mờ trên $Y \times Z$. Hợp thành $R_1 \circ R_2$ của R_1, R_2 là quan hệ mờ trên $X \times Z$.

a) *Hợp thành max-min* (max–min composition) được xác định bởi

$$\mu_{R_1 \circ R_2}(x,z) = \max_y\{\min(\mu_{R_1}(x,y), \mu_{R_2}(y,z)\}, \ \forall (x,z) \in X \times Z.$$

b) *Hợp thành max-prod* cho bởi

$$\mu_{R_1 \circ R_2}(x,z) = \max_y\{(\mu_{R_1}(x,y) \cdot \mu_{R_2}(y,z)\}, \ \forall (x,z) \in X \times Z.$$

c) *Hợp thành max–** được xác định bởi toán tử $*: [0,1]^2 \to [0,1]$

$$\mu_{R_1 \circ R_2}(x,z) = \max_y\{\mu_{R_1}(x,y) * \mu_{R_2}(y,z)\}, \ \forall (x,z) \in X \times Z.$$

Ví dụ 3.1: Cho $X=\{x_1,x_2,x_3\}$, $Y=\{y_1,y_2\}$, $Z=\{z_1,z_2\}$, với quan hệ mờ R cho trên $X \times Y$, S là quan hệ mờ cho trên $Y \times Z$ cho bởi ma trận

$$R = \begin{bmatrix} 0.8 & 1 \\ 1 & 0.7 \\ 0.2 & 0.6 \end{bmatrix}, \qquad S = \begin{bmatrix} 0.7 & 1 \\ 0.6 & 0.9 \end{bmatrix}$$

thì hợp thành max–min là $S \circ R = \begin{bmatrix} 0.7 & 0.9 \\ 0.7 & 1 \\ 0.6 & 0.6 \end{bmatrix}$ và max–prod là $S \circ R = \begin{bmatrix} 0.6 & 0.9 \\ 0.7 & 1 \\ 0.36 & 0.54 \end{bmatrix}$

Bây giờ chúng ta xét một số quan hệ nhị nguyên mờ trên cùng một không gian nền.

Giả thiết T là t-chuẩn.

Định nghĩa 3.6: Cho R_1, R_2 là quan hệ mờ trên $X \times X$, phép T–hợp thành cho một quan hệ $R_1 \circ_T R_2$ trên $X \times X$ xác định bởi

$$R_1 \circ_T R_2(x,z) = \sup_{y \in X} T(R_1(x,y), R_2(y,z)).$$

Định lý 3.1: Cho R_1, R_2, R_3 là những quan hệ mờ trên $X \times X$, khi đó:

a) $R_1 \circ_T (R_2 \circ_T R_3) = (R_1 \circ_T R_2) \circ_T R_3$

b) Nếu $R_1 \subseteq R_2$ thì $R_1 \circ_T R_3 \subseteq R_2 \circ_T R_3$ và $R_3 \circ_T R_1 \subseteq R_3 \circ_T R_2$

<u>*Tính bắc cầu*</u>:

Định nghĩa 3.7: Quan hệ mờ R trên $X \times X$ gọi là:

a) *min–chuyển tiếp* nếu $\min\{R(x,y), R(y,z)\} \le R(x,z) \quad \forall x,y,z \in X$.

b) *bắc cầu yếu* nếu $\forall x,y,z \in X$ có

$$R(x,y) > R(y,x) \text{ và } R(y,z) > R(z,y) \text{ thì } R(x,z) > R(z,x).$$

c) *bắc cầu tham số* nếu có một số $0<\theta<1$ sao cho: Nếu $R(x,y) > \theta > R(y,x)$ và $R(y,z) > \theta > R(z,y)$ thì $R(x,z) > \theta > R(z,x)$ $\forall x,y,z \in X$.

Định lý 3.1: (xem [7])

a) Nếu R là quan hệ mờ có tính chất *min–bắc cầu* thì R là quan hệ mờ có tính chất *bắc cầu tham số* với mọi $0< \theta< 1$.

b) Nếu R là quan hệ mờ có tính chất *bắc cầu tham số* thì R là quan hệ mờ có tính chất *bắc cầu yếu*.

1.3.2 Phương trình quan hệ mờ

Phương trình quan hệ mờ lần đầu tiên nghiên cứu bởi GS. Sanchez năm 1976, đóng vai trò quan trọng trong các lĩnh vực phân tích các hệ mờ, thiết kế các bộ điều khiển mờ, quá trình lấy quyết định và nhận dạng mờ.

Dạng đơn giản nhất có thể diễn đạt như sau:

Cho một hệ mờ biểu diễn dưới dạng một quan hệ mờ nhị nguyên R trên không gian tích $X\times Y$. Đầu vào (input) của hệ là một tập mờ A cho trên không gian nền input X. Tác động của đầu vào A với hệ R sẽ là phép hợp thành $A\circ R$ sẽ cho ở đầu ra (output) một tập mờ trên không gian nền Y, kí hiệu là B. Khi ấy chúng ta có $A\circ R = B$.

Nếu chúng ta sử dụng phép hợp thành max–min thì hàm thuộc của B cho bởi

$$\mu_B(y = \mu_{A\circ R}(y) = \max_x(\min_y[\mu_A(x), \mu_R(x,y)])$$

Ví dụ 3.2: Cho input là tập mờ A trên X và quan hệ mờ R trên $X\times Y$ như sau:

$X=\{x_1,x_2,x_3\}$, $Y=\{y_1,y_2,y_3\}$,

$$A = (0{,}2/x_1 \quad 0{,}8/x_2 \quad 1/x_3) = (0{,}2 \quad 0{,}8 \quad 1)$$

$$A\circ R = \begin{bmatrix} 0.7 & 1 & 0.4 \\ 0.5 & 0.9 & 0.6 \\ 0.2 & 0.6 & 0.3 \end{bmatrix}.$$

Khi đó chúng ta có

$$B = A\circ R = (0.2 \quad 0.8 \quad 1)\circ \begin{bmatrix} 0.7 & 1 & 0.4 \\ 0.5 & 0.9 & 0.6 \\ 0.2 & 0.6 & 0.3 \end{bmatrix} = (0{,}5 \quad 0{,}8 \quad 0{,}6) = \frac{0{,}5}{y_1} + \frac{0{,}8}{y_2} + \frac{0{,}6}{y_3}.$$

1.4 Suy luận xấp xỉ và suy diễn mờ

1.4.1 Chúng ta sẽ trình bày đủ đơn giản vấn đề suy luận xấp xỉ dưới dạng những mệnh đề với các biến ngôn ngữ như đời thường vẫn dùng như: "máy lạnh", "ga yếu", hay những quy tắc, những luật dạng mệnh đề "nếu quay tay ga mạnh thì tốc độ xe sẽ nhanh".

Suy luận xấp xỉ - hay còn gọi là suy luận mờ - đó là quá trình suy ra những kết luận dưới dạng các mệnh đề mờ trong điều kiện các quy tắc, các luật, các dữ liệu đầu vào cho trước cũng không hoàn toàn xác định. Chúng ta sẽ hạn chế bởi những luật đơn giản như dạng *modus ponens* hay *modus tollens* đã nêu ở phần đầu.

Trước tiên chúng ta nhớ lại trong giải tích toán học đã dùng quá trình lập luận sau:

Định lý:	Nếu một hàm số là khả vi thì nó liên tục
Sự kiện:	Hàm f khả vi
Kết luận:	f liên tục

đây là dạng suy luận dựa vào luật *modus ponens*. Bây giờ ta tìm cách diễn đạt cách suy luận quen thuộc trên dưới dạng sao cho có thể suy rộng cho logic mờ.

Ký hiệu: U = không gian nền = không gian tất cả các hàm số.

Ví dụ đơn giản có thể hiểu

$U = \{g : R \rightarrow R\}$,

A = {các hàm khả vi},

B = {các hàm liên tục}.

Hãy chọn hai mệnh đề P = "$g \in A$" và Q = "$g \in B$". Khi ấy chúng ta có

Luật (tri thức):	$g \Rightarrow B$
Sự kiện:	P đúng (true)
Kết luận:	Q đúng (true)

ở đây chúng ta đã sử dụng luật *modus ponens* $((P \Rightarrow Q) \wedge P) \Rightarrow Q$.

1.4.2 Bây giờ đã có thể chuyển sang suy diễn mờ cùng dạng.

Luật mờ:	Nếu góc tay quay ga lớn thì xe đi nhanh
Sự kiện mờ:	Góc tay ga quay khá lớn
Hệ quả:	Xe đi khá nhanh

Zadeh đã diễn đạt sự kiện trên bằng các biến ngôn ngữ: góc tay quay, tốc độ, nhiệt độ, áp lực, tuổi tác và các mệnh đề mờ dạng tương ứng. Chúng ta làm rõ cách tiếp cận của Zadeh qua vài ví dụ.

1.4.2.1 Biến ngôn ngữ

Ví dụ 3.3: Ta nói *"Nam có tuổi trung niên"*, khi ấy chọn

x = biến ngôn ngữ "*Tuổi*",

không gian nền là thời gian sống

U = [0, 130 năm],

A= tập mờ "*trung niên*".

Một cách tự nhiên, ta gán cho A là một tập mờ trên U với hàm thuộc $A(u) : U \rightarrow [0,1]$.

Sự kiện "*có thể tuổi của Nam là* 40" dĩ nhiên không chắc chắn và khá hợp lý nếu diễn đạt như một khả năng, trong [4,5] Zadeh đề nghị

Khả năng (Tuổi của Nam = 40) = Poss(x = 40)
= độ thuộc của số 40 vào tập mờ $A = A(40)$.

Mệnh đề mờ

"*Nam có tuổi trung niên*"

bây giờ được diễn đạt thành mệnh đề

$P = \{ x = A \}$ = {biến x nhận giá trị mờ A trên không gian nền U}
= {x is A } (theo dạng tiếng Anh).

1.4.2.2 **Ví dụ 3.4**: Đối với suy luận mờ cho ở đầu mục này chúng ta có thể dùng biến ngôn ngữ

x= "góc tay quay"

trên không gian nền $U = [0,360\,°]$ (cho phép quay tay ga của xe máy). A = ""góc lớn" là một tập mờ trên U (trong trường hợp này tiện hơn dùng khái niệm số mờ A), với hàm thuộc $A(u)$: $U \to [0,1]$.

Tương tự, biến ngôn ngữ y = "tốc độ xe", với không gian nền

V = [0 km/giờ, 150 km/giờ].

Q = "xe đi nhanh"= một tập mờ B trên không gian nền V với hàm thuộc $B(v) : V \to [0,1]$. Khi ấy

P = "góc tay quay lớn" = { $x = A$ } (x is u).

Q = "xe đi nhanh" = { $y = B$ }.

và luật mờ có dạng $P \Rightarrow Q$.

Như vậy một luật mờ dạng "If P then Q" sẽ được biểu diễn thành một quan hệ mờ R của phép kéo theo $P \Rightarrow Q$ với hàm thuộc của R trên không gian nền $U\times V$ được cho bởi phép kéo theo mà bạn dự định sử dụng:

$$R_{(A,B)}(u, v) = R_{P\Rightarrow Q}(u, v) = I(A(u), B(v)), \quad \text{với mọi } (u, v) \in U\times V.$$

Bây giờ quy trình suy diễn mờ đã có thể xác định:

Luật mờ (tri thức):	$P \Rightarrow Q$, với quan hệ cho bởi $I(A(u),B(v))$
Sự kiện mờ (đầu vào):	$P' = \{x = A'\}$, xác định bởi tập mờ A' trên U
Kết luận:	$Q' = \{y = B'\}$

Sau khi đã chọn phép kéo theo I xác định quan hệ mờ $R_{(A,B)}$, B' là một tập mờ trên V với hàm thuộc của B' được tính bằng phép hợp thành $B' = A' \circ R_{(A,B)}$, cho bởi công thức:

$$B'(v) = \max_{u\in U}\{\min(A'(u), I(A(u), B(v)))\}, \text{ với mỗi } v \in V.$$

1.4.3 Tiếp tục cách biểu diễn và diễn đạt như vậy, ta có thể xét dạng

"If P then Q else Q_1"

quen biết trong logic cổ điển và thường hay sử dụng trong các ngôn ngữ lập trình của ngành Tin học.

Có thể chọn những cách khác nhau diễn đạt mệnh đề này, sau đấy tìm hàm thuộc của biểu thức tương ứng. Chẳng hạn, chúng ta chọn

"If P then Q else Q_1" = $(P \wedge Q) \vee (\neg P \wedge Q_1)$.

Thông thường Q và Q_1 là những mệnh đề trong cùng một không gian nền.

Giả thiết Q và Q_1 được biểu diễn bằng các tập mờ B và B_1 trên cùng không gian nền V, với các hàm thuộc tương ứng $B : V \rightarrow [0,1]$ và $B_1 : V \rightarrow [0,1]$. Nếu Q và Q_1 không cùng không gian nền thì cũng sẽ xử lý tương tự nhưng với công thức phức tạp hơn.

• Kí hiệu $R(P, Q, Q') = R(A, B, B_1)$ là quan hệ mờ trên $U\times V$ với hàm thuộc cho bởi biểu thức

$R(u,v) = \max\{\min(A(u), B(v)), \min(1-A(u), B_1(v))\}$, với mọi $(u,v) \in U\times V$.

Tiếp tục quy trình này chúng ta có thể xét những quy tắc lấy quyết định phức tạp hơn. Chẳng hạn chúng ta xét một quy tắc trong hệ thống mờ có hai biến đầu vào và một đầu ra dạng

If A_1 and B_1 then C_1

else If A_2 and B_2 then C_2

else ...

1.4.4 Một dạng suy rộng khác trong cơ sở tri thức của nhiều hệ mờ thực tiễn, ví dụ điển hình là trong các hệ điều khiển mờ, có thể phát biểu dưới dạng sau:

Cho $x_1, x_2, \ldots, x_m$ là các biến vào của hệ thống, y là biến ra. Các tập A_{ij}, B_j, với $i = 1, \ldots, m$, $j = 1, \ldots, n$ là các tập mờ trong các không gian nền tương ứng của các biến vào và biến ra đang sử dụng của hệ thống, các R_j là các suy diễn mờ (các luật mờ) dạng "Nếu ... thì ..." (dạng if ... then)

R_1: Nếu x_1 là $A_{1,1}$ và ... và x_m là $A_{m,1}$ thì y là B_1

R_2: Nếu x_1 là $A_{1,2}$ và ... và x_m là $A_{m,2}$ thì y là B_2

...

R_n: Nếu x_1 là $A_{1,n}$ và ... và x_m là $A_{m,n}$ thì y là B_n

Bài toán

Cho:	Nếu x_1 là e_1* và ... và x_m là e_m*
Tính:	Giá trị y là u*

ở đây e_1*, ... , e_m* là các giá trị đầu vào hay sự kiện (có thể mờ hoặc giá trị rõ).

Chúng ta có thể nhận thấy rằng *phần cốt lõi của nhiều hệ mờ cho bởi cơ sở tri thức dạng R={các luật R_i} và các cơ chế suy diễn cài đặt trong mô tơ suy diễn.*

Tính toán quan hệ mờ cho những bộ luật phức tạp như thế các bạn có thể xem thêm công trình của M. Mizumoto và H.J. Zimmermann. Những kiến thức về suy diễn mờ liên quan tới lập luận ngôn ngữ có thể đọc thêm chương 2 của Nguyễn Cát Hồ .

1.5 Ví dụ bằng số

1.5.1 Để minh hoạ trong phần cuối chúng ta xét ví dụ bằng số trực tiếp trích từ [7]. Ví dụ chúng ta nghe thấy câu nói: "Nếu nhiệt độ của hệ thống lạnh, thì áp suất của hệ thống yếu". Rõ ràng đây là một luật mờ dạng $P \Rightarrow Q$.

Trước tiên chúng ta chọn không gian nền với các trạng thái cơ sở. Ví dụ:

U ={nhiệt độ của hệ thống} = { thấp, trung bình thấp, hơn trung bình, cao}
$= \{u_1, u_2, u_3, u_4\}$.

V ={áp suất của hệ thống} = {thấp, trung bình thấp, trung bình, hơn trung bình, cao}
$= \{v_1, v_2, v_3, v_4, v_5\}$.

Trong trường hợp này mỗi mệnh đề A_1 trên U có hàm thuộc hoàn toàn xác định bởi vectơ { $A_1(u)$: $u \in U$ }. Như vậy, chẳng hạn

Tập mờ A_1 biểu diễn mệnh đề: "nhiệt độ lạnh" = {1 0.6 0 0},

Tập mờ B_1 biểu diễn mệnh đề: "áp suất thấp" = {1 0.8 0.1 0 0}.

Để tính độ thuộc của quan hệ mờ, người ta thác triển A_1 lên không gian nền $U \times V$. Khi ấy hàm thuộc của A_1 sẽ kí hiệu $\text{ext}_{U\times V}\, A_1$ có dạng

$$\text{ext}_{U\times V}\, A_1 = \begin{vmatrix} 1 & 1 & 1 & 1 & 1 \\ 0.6 & 0.6 & 0.6 & 0.6 & 0.6 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{vmatrix}.$$

Do $P \Rightarrow Q$ là đồng nhất với biểu thức $\neg A_1 \vee (A_1 \wedge B_1)$, cho nên để tính hàm thuộc xác định trên $U \times V$ của quan hệ này chỉ cần tính ma trận

$$(\text{ext}_{U\times V} \neg A_1) \vee ((\text{ext}_{U\times V}\, A_1) \wedge (\text{ext}_{U\times V} B_1))$$

Sau đây là các ma trận tương ứng:

$$\text{ext}_{U\times V}\ \neg A_1 = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 \\ 0.4 & 0.4 & 0.4 & 0.4 & 0.5 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{vmatrix},$$

$$(\text{ext}_{U\times V}\ A_1)\ \wedge (\text{ext}_{U\times V} B_1) = \begin{vmatrix} 1 & 0.8 & 0.1 & 0 & 0 \\ 0.6 & 0.6 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{vmatrix}.$$

Biểu diễn và tính toán như trên vì chúng ta sẽ tính quan hệ $R(A_1, B_1)$ theo phép kéo theo $I_S(u, v)$ như trong ví dụ a phần 2.4.5. Kết quả thu được quan hệ

$$R_{P\Rightarrow Q} = R(A_1, B_1) = \begin{vmatrix} 1 & 0.8 & 0.1 & 0 & 0 \\ 0.6 & 0.6 & 0.4 & 0.4 & 0.4 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{vmatrix}.$$

Đây chính là quan hệ mờ biểu thị quan hệ $P \Rightarrow Q$, thông qua các biến ngôn ngữ "nhiệt độ, áp suất" và các tập mờ A_1, B_1 tương ứng.

Tiếp tục, chúng ta có thể tiến hành các suy diễn mờ. Chẳng hạn, sự kiện đầu vào quan sát được là "*nhiệt độ của hệ thống hơi lạnh*". Mệnh đề mờ P' này diễn đạt qua tập mờ

P' với hàm thuộc trên không gian nền U cho bởi vector $A' = \{0.8 \quad 1 \quad 0.3 \quad 0\}$.

Như thế chúng ta có quá trình suy diễn mờ:

Luật mờ (tri thức):	$R(A_1, B_1)$
Sự kiện mờ (đầu vào):	A'
Kết luận:	B'

B' là một tập mờ trên V được tính bằng phép hợp thành $B' = R(A_1, B_1) \circ A'$.

Áp dụng vào trường hợp này chúng ta thu được hàm thuộc của B' là vectơ

$B' = \{0.8 \quad 0.8 \quad 0.4 \quad 0.4\}$.

Nếu bây giờ quan hệ $R(A_1, B_1)$ được tính theo phép kéo theo $I_S(u, v)$ như trong ví dụ 6.8.b thì

$$R_{P\Rightarrow Q} = R(A_1, B_1) = \begin{vmatrix} 1 & 0.8 & 0.1 & 0 & 0 \\ 1 & 0.8 & 0.4 & 0.4 & 0.4 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{vmatrix}$$

và kết quả của suy diễn mờ trong trường hợp này sẽ là $B' = R(A_1,B_1) \circ A'$ là tập mờ kết luận (kết quả đầu ra) trên không gian nền V có hàm thuộc cho bởi vector

$$B' = \{1 \quad 0.8 \quad 0.4 \quad 0.4 \quad 0.4\}.$$

1.5.2 Tính toán minh họa cho mệnh đề dạng " If P then Q else Q1".

Giả sử mệnh đề Q_1 cùng không gian nền V với mệnh đề Q, chẳng hạn Q_1 = "áp suất của hệ thống trung bình". Q_1 sẽ được biểu diễn qua biến ngôn ngữ "áp suất" và tập mờ B_2 với hàm thuộc cho trên V là vector

$$B_2 = \{0 \quad 0.6 \quad 1 \quad 0.6 \quad 0\}.$$

Khi ấy

$$\text{ext}_{U\times V} B_2 = \begin{vmatrix} 0 & 0.6 & 1 & 0.6 & 0 \\ 0 & 0.6 & 1 & 0.6 & 0 \\ 0 & 0.6 & 1 & 0.6 & 0 \\ 0 & 0.6 & 1 & 0.6 & 0 \end{vmatrix},$$

$$(\text{ext}_{U\times V} \rceil A_1) \wedge (\text{ext}_{U\times V} B_2) = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0.4 & 0.4 & 0.4 & 0 \\ 0 & 0.6 & 1 & 0.6 & 0 \\ 0 & 0.6 & 1 & 0.6 & 0 \end{vmatrix}.$$

Cuối cùng thu được

$$R(\text{If } P \text{ then } Q \text{ else } Q_1) = \begin{vmatrix} 1 & 0.8 & 0.1 & 0 & 0 \\ 0.6 & 0.6 & 0.4 & 0.4 & 0 \\ 0 & 0.6 & 1 & 0.6 & 0 \\ 0 & 0.6 & 1 & 0.6 & 0 \end{vmatrix}.$$

1.6 Sự phát triển của công nghệ mờ

Do hạn chế về thời gian, người viết tập trung trình bày về một vài nét về tình hình tại Nhật Bản. Trong quá trình phát triển của Lý thuyết tập mờ và công nghệ mờ tại Nhật Bản phải nhắc tới dự án lớn LIFE (the Laboratory for International Fuzzy Engineering) 1989–1995 do G.S. T.Terano (Tokyo Institute of Technology) làm Giám đốc điều hành – theo sáng kiến và sự tài trợ chính của Bộ ngoại thương và công nghiệp Nhật Bản. Phòng thí nghiệm LIFE được thiết kế bởi G.S. M. Sugeno. Chính Giáo sư cũng đã thuyết phục được nhiều công ty công nghiệp hàng đầu của Nhật Bản cung cấp tài chính và nhân lực, trở thành thành viên tập thể của dự án và chính họ trực tiếp biến các sản phẩm của phòng thí nghiệm thành sản phẩm hàng hoá.

Và kết quả là, theo Datapro, nền công nghiệp sử dụng công nghệ mờ của Nhật Bản, năm 1993 có tổng doanh thu khoảng 650 triệu USD, thì tới năm 1997 đã ước lượng cỡ 6,1 tỷ

USD và hiện nay hàng năm nền công nghiệp Nhật Bản chi 500 triệu USD cho nghiên cứu và phát triển lý thuyết mờ và công nghệ mờ. Theo Giáo sư T. Terano [6] quá trình phát triển của công nghệ mờ có thể chia thành bốn giai đoạn sau:

1) *Giai đoạn* 1: Lợi dụng tri thức ở mức thấp.

Thực chất: Những ứng dụng trong công nghiệp chủ yếu là biểu diễn tri thức định lượng của con người.

Ví dụ điển hình: Điều khiển mờ.

Trong giai đoạn ban đầu này, chủ yếu là cố gắng làm cho máy tính hiểu một số từ định lượng của con người vẫn quen dùng (như 'cao, nóng, ấm, yếu', v.v.). Một lí do rất đơn giản để đi tới phát triển điều khiển mờ là câu hỏi sau: "*Tại sao các máy móc đơn giản trong gia đình ai cũng điều khiển được mà máy tính lại không điều khiển được ?*".

Có thể hầu hết các hệ điều khiển mờ là ở mức này. Thực tế tại mức ban đầu này đã đưa vào sử dụng rất nhiều loại máy mới có sử dụng logic mờ. *Đó là sự kiện rất quan trọng trong quá trình phát triển của logic mờ*, nhưng đó vẫn là các hệ thuộc giai đoạn 1.

2) *Giai đoạn* 2: Sử dụng tri thức ở mức cao.

Thực chất: Dùng logic mờ để biểu diễn tri thức.

Ví dụ: – Các hệ chuyên gia mờ.

– Các ứng dụng ngoài công nghiệp: y học, nông nghiệp, quản lý, xã hội học, môi trường.

Trong giai đoạn này cố gắng trang bị cho máy tính những tri thức cơ bản và sâu sắc hơn, những tri thức định tính mà trước tới nay chưa thể biểu diễn bằng định lượng, ví dụ như trong các hệ chuyên gia mờ, mô hình hoá nhiều bài toán khó trong quản lý các nhà máy mà trước đây chưa làm được.

3) *Giai đoạn* 3: Liên lạc–giao tiếp.

Thực chất: Giao lưu giữa người và máy tính thông qua ngôn ngữ tự nhiên.

Ví dụ: – Các robot thông minh.

– Các hệ hỗ trợ quyết định dạng đối thoại.

4) *Giai đoạn* 4: Trí tuệ nhân tạo tích hợp.

Thực chất: Giao lưu và tích hợp giữa trí tuệ nhân tạo, logic mờ, mạng nơron và con người.

Ví dụ: – Giao lưu con người và máy tính.

– Các máy dịch thuật.

– Các hệ hỗ trợ lao động sáng tạo.

Giáo sư Terano còn cho rằng sự phát triển của công nghệ mờ và các hệ mờ tại Nhật Bản đã và sẽ đi qua bốn giai đoạn trên.

Một số thông tin khác người đọc có thể tham khảo thêm các tài liệu [3,4.5.6] và các tài trích dẫn trong đó. Về các ứng dụng của công nghệ mờ bạn đọc có thể tham khảo thêm tài liệu [8].

2 Mạng nơron nhân tạo và hệ mờ

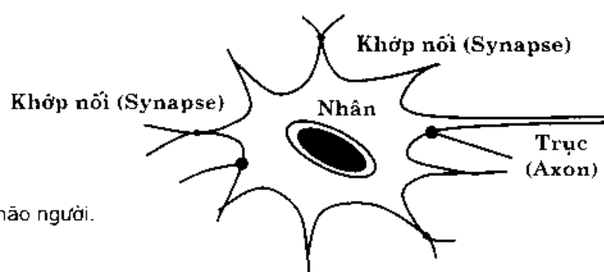
2.1 Mạng nơron nhân tạo

2.1.1 Não và nơron sinh học

Não là tổ chức vật chất cao cấp, có cấu tạo vô cùng phức tạp, dày đặc các mối liên kết giữa các nơron nhưng xử lý thông tin rất linh hoạt trong một môi trường bất định.

Trong bộ não có khoảng 10^{11} - 10^{12} nơron và mỗi nơron có thể liên kết với 10^{4} nơron khác qua các khớp nối. Những kích hoạt hoặc ức chế này được truyền qua trục nơron (axon) đến các nơron khác.

Trên hình 3 là hình ảnh của tế bào nơron trong não con người.



Hình 3: Cấu tạo của nơron của não người.

Khi người ta nhìn não từ góc độ tính toán, chúng ta dễ dàng phát hiện cách thức tính toán của não khác xa với tính toán theo thuật toán và chương trình chúng ta thường làm với sự trợ giúp của máy tính.

Sự khác biệt cơ bản trước tiên là ở 2 điểm rất quan trọng sau:

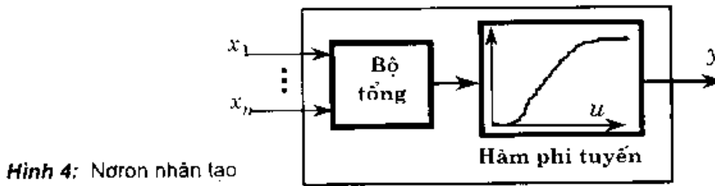
- Quá trình tính toán được tiến hành song song và phân tán trên nhiều nơron gần như đồng thời.
- Tính toán thực chất là quá trình học, chứ không phải theo sơ đồ định sẵn từ trước.

2.1.2 Mạng nơron nhân tạo

2.1.2.1 Nơron nhân tạo

Khai thác nhận xét trên, bắt chước não, các nhà khoa học đã có mô hình tính toán mới: đó là các mạng nơron nhân tạo (*Artifical Neural Networks ANN*).

Một nơron nhân tạo (một đơn vị xử lý – PE) phản ảnh các tính chất cơ bản của nơron sinh học và được mô phỏng dưới dạng như hình 4.



Hình 4: Nơron nhân tạo

Đầu vào của nơron nhân tạo gồm n tín hiệu x_i (i = 1,2, ... , n). Đầu ra là tín hiệu y. Trạng thái bên trong của nơron được xác định qua bộ tổng các đầu vào có trọng số w_i (i = 1,2, ... , n). Đầu ra y của nơron được xác định qua hàm phi tuyến nào đó f.

Như vậy mô hình định lượng của nơron nhân tạo như sau:

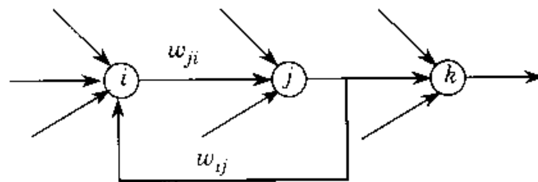
$$y(t) = f\left(\sum_{i=1}^{n} w_i x_i(t) - \theta\right).$$

ở đây $net = \sum_{i=1}^{n} w_i x_i(t) = I(t)$ là tín hiệu tổng hợp đầu vào, w_i – các trọng số, i = 1,2, ... , n đặc trưng cho tính liên kết của các khớp synap. θ–ngưỡng kích hoạt nơron, t–thời gian. n–số tín hiệu đầu vào, f–hàm kích hoạt.

Do vậy người ta rất hay dùng kí hiệu sau: Đầu ra $out = y(t) = f(net)$.

Tóm lại có thể xem *nơron là một hàm phi tuyến nhiều đầu vào, một đầu ra.*

Các nơron có thể liên kết với nhau tạo thành mạng nơron nhân tạo. Ví dụ nơron i liên kết với nơron j theo hai chiều thuận nghịch (có thông tin phản hồi) như ở hình 5.



Hình 5: Liên kết 2 chiều giữa nơron i và nơron j.

2.1.2.2 Các giai đoạn phát triển của mạng nơron

Theo R. Schalkoff [11] thì có thể chia sự phát triển của mạng nơron nhân tạo thành 3 giai đoạn:

1) *Giai đoạn 1:* Tiền Perceptron (những năm 1940 – những năm 1960)

Dễ dàng nhận thấy rằng trong giai đoạn này mạng chưa đủ phức tạp cho nên chưa có khả năng giải quyết các bài toán khó có sức thuyết phục.

Trong giai đoạn này cần nhắc tới các sự kiện sau:

- Lần đầu tiên McCulloch và Pitts,1943 giới thiệu mô hình toán học của mạng nơron.
- Rosenblatt, 1957 định nghĩa Perceptron, mong muốn khẳng định các nơron liên kết, phi tuyến tạo nên mạng thích nghi có thể góp phần giải quyết các bài toán nhận dạng.
- 1960 Windrow đóng góp chính là thuật toán trung bình bình phương bé nhất (LMS) cho mô hình Adaline/Madaline.
- Kết quả của Minsky và Papert 1969.

2) *Giai đoạn 2*: Hậu Perceptron

Trong giai đoạn này mô hình perceptron được phát huy với những thuật học truyền thẳng và liên kết suy rộng. Đã tìm thêm nhiều cấu trúc mới, ứng dụng mới, trong đó cần kể tới:

- Mạng truyền thẳng với thuật toán lan truyền ngược (luật Delta suy rộng–GDR), 1985.
- Mạng dùng các hàm cơ sở xuyên tâm (mạng RBF).
- Các mạng Hopfield hồi quy, 1982.
- Bộ nhớ liên hợp hai chiều (BAM), 1987.
- Làm sâu sắc hơn nhiều khái niệm và các thuật toán đã có.
- Công trình về các mạng thích nghi của Grossberg và Kohonen.

3) *Giai đoạn 3*: Gần đây và hiện nay

Tiếp tục suy rộng và đưa vào thực tiễn nhiều mô hình và thuật toán đã hoàn chỉnh hơn. Những vấn đề chính hiện nay phải làm là:

- Đánh giá xác thực những hạn chế của mạng nơron.
- Các khả năng suy rộng khác nhau.
- Phối hợp công nghệ mạng nơron với các công nghệ của logic mờ và các thuật toán di truyền.
- Cài đặt các mạng nơron nhân tạo bằng các phần cứng chuyên dụng.

2.1.3 Sức mạnh của mô hình mạng nơron

Những mô hình mạng nơron đã trình bày có tiềm năng tạo nên một cuộc cách mạng trong công nghệ máy tính và các quá trình xử lý thông tin. Những mong muốn và hy vọng đó chủ yếu bắt nguồn từ các đặc trưng chính sau:

a) Khả năng của *các quá trình xử lý song song và phân tán*: Có thể đưa vào mạng một lượng lớn các nơron liên kết với nhau theo những lược đồ với các kiến trúc khác nhau.

b) Khả năng *thích nghi và tự tổ chức*: về đặc trưng này người ta đề cập tới khả năng xử lý thích nghi và điều chỉnh bền vững dựa vào các thuật toán học thích nghi và các quy tắc tự tổ chức.

c) Khả năng *dung thứ lỗi*: Cố gắng bắt chước khả năng dung thứ lỗi của não theo nghĩa hệ thống có thể tiếp tục làm việc và điều chỉnh khi nhận tín hiệu vào một phần thông tin bị sai lệch hoặc bị thiếu.

d) *Xử lý các quá trình phi tuyến*: Đặc trưng này rất quan trọng, ví dụ trong xấp xỉ mạng, miễn nhiễu (chấp nhận nhiễu) và có khả năng phân lớp.

2.1.4 Các phạm vi ứng dụng

Lĩnh vực ứng dụng của mạng nơron nhân tạo rất rộng, chủ yếu trong các vùng sau:

1) *Lĩnh vực* 1: *Phân lớp* (classification), *tách cụm* (clustering), *dự đoán* (diagnoisis) và *liên kết*. Có thể đây là lĩnh vực tìm thấy nhiều ứng dụng nhất và cũng được nghiên cứu sôi động nhất. Nhóm mô hình này nhận những tín hiệu vào tĩnh hoặc tín hiệu theo thời gian và cần nhận dạng hoặc phân lớp chúng. Thuật toán phân lớp cần huấn luyện mạng sao cho khi tín hiệu vào bị biến dạng ít nhiều thì mạng vẫn nhận đúng dạng thực của chúng. Mạng cần có khả năng miễn nhiễu tốt, bởi vì đây là một mong muốn tha thiết của nhiều ứng dụng.

2) *Lĩnh vực* 2: *Các bài toán tối ưu*.Vấn đề chính ở đây là tìm những thuật toán huấn luyện mạng sao cho góp phần tìm nghiệm cho nhiều lớp bài toán tối ưu toàn cục.Trong nhóm các thuật toán ứng dụng mạng nơron, người ta đã quan tâm tới sự kết hợp mạng nơron với các thuật toán di truyền.

3) *Lĩnh vực* 3: *Hồi quy và tổng quát hoá* (Regression and Generalization). Trước đây các bài toán hồi quy đã được tích cực nghiên cứu. Qua hồi quy tuyến tính và phi tuyến người ta gắng tìm các đường thẳng hoặc các đường hồi quy phi tuyến trơn sao cho khớp với mẫu. Trong các bài toán hồi quy người ta thường dùng các thuật học có giám sát. Bài toán suy rộng khó hơn, vì dữ liệu được học mới chỉ có một phần.

4) *Lĩnh vực* 4: *Hoàn chỉnh dạng* (Pattern completion). Bài toán là hoàn chỉnh "đủ" dữ liệu ban đầu sau khi đã bị mất đi một phần (hay ta chỉ thu được một phần). Người ta đã quan tâm tới hai lớp mô hình: Mô hình Markov và các mạng có độ trễ với các mạng nơron nhiều lớp, máy Bolzmann và mạng Hopfield tĩnh.

Trong những phần còn lại của phần tổng quan này, chúng ta sẽ xét thêm về cấu trúc của mạnh, các hàm kích hoạt và bài toán huấn luyện mạng.

2.1.5 Cấu trúc mạng nơron

Cấu trúc của mạng nơron chủ yếu được đặc trưng bởi loại của các nơron và mối liên hệ xử lý thông tin giữa chúng.

Về cấu trúc của nơron: Chủ yếu người ta quan tâm tới cách "tổng" các tín hiệu vào, ngưỡng tại mỗi nơron và các hàm chuyển – hàm kích hoạt. Sau đây là một số hàm kích hoạt.

2.1.5.1 Hàm kích hoạt

Hàm kích hoạt của từng nơron trong mạng nơron đóng vai trò quan trọng trong sự liên kết giữa các nơron. Hàm này đặc trưng cho mức độ liên kết giữa các nơron. Trong lý thuyết mạng nơron, phép tổng hợp các tín hiệu đầu vào và thường được ký hiệu dưới dạng sau: ví dụ đối với nơron j có m tín hiệu đầu vào x_i

$$net_j = \sum_{i=1}^{m} x_i w_{ji} \quad ; \quad w_{ji} = (w_{j1}, \dots, w_{jm})$$

Đầu ra của nơron j thường ký hiệu là out_j hoặc f_j. Sau đây là vài dạng hàm kích hoạt:

a) $$f_j = out_j = \begin{cases} 1, & \text{nếu} \quad (net_j - \theta_j) \geq 0 \\ -1 & \text{nếu} \quad (net_j - \theta_j) < 0 \end{cases}.$$

b) Dạng hàm Gauss: $f_j = out_j = \exp(-(net_j - \theta_j)^2)$.

c) Dạng hàm sigmoid (hay hàm logistic): $f_j = out_j = (1 + \exp(-(net_j - \theta_j))) - 1$.

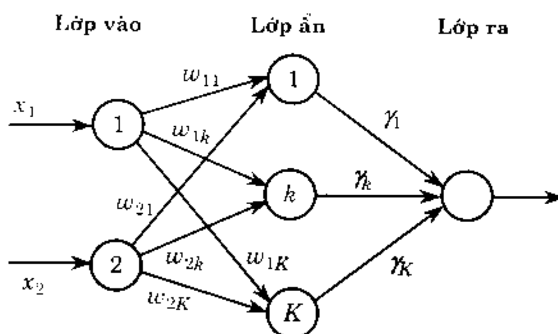
Các loại hàm kích hoạt còn có nhiều biến thể khác nhau.

2.1.5.2 Liên kết mạng

Sự liên kết trong mạng nơron tuỳ thuộc vào nguyên lý tương tác giữa đầu ra của từng nơron riêng biệt với các nơron khác và tạo ra cấu trúc mạng nơron. Về nguyên tắc sẽ có rất nhiều kiểu liên kết giữa các nơron. Mỗi nơron là một nút của mạng. Một số cấu trúc hay gặp trong những ứng dụng có dạng sau.

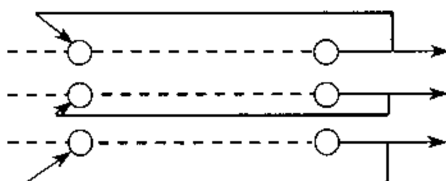
1) *Các mạng truyền thẳng* (Feedforward neural networks).

Đó là một đồ thị định hướng hữu hạn không chu trình (acyclid). Mỗi nút là một nơron, có phân biệt nút vào và nút ra. Các nơron chia theo lớp (layers). Trên mỗi cung có trọng số w_{ij} nối nơron j với nơron i trong lớp sau. Mỗi nút k không phải nút vào có gán ngưỡng θ_k. Mạng trình bày trên hình 6 sẽ minh hoạ cho một mạng nơron truyền thẳng có ba lớp.



Hình 6: Cấu trúc truyền thẳng phân lớp.

2) *Mạng có nối ngược*: Các mạng có thông tin và xử lý theo 2 chiều (hình 7, có nối ngược–còn được gọi là mạng hồi quy recurrent networks).



Hình 7: Mạng Hopfield, dùng liên kết phản hồi

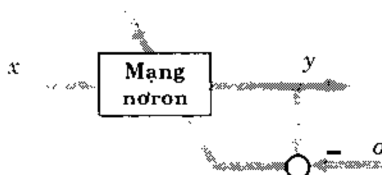
Cấu trúc của các mạng nơron có thể tham khảo thêm trong phần 2 về các lớp mạng nơron cơ bản.

2.1.6 Bài toán huấn luyện mạng

Bài toán huấn luyện mạng (hay bài toán học – the problem of learning) là xác định (nhận dạng – to identify) các tham số của mạng, chủ yếu là các trọng số liên kết mạng và cấu trúc – các dạng liên kết của các nơron, giữa các lớp dựa trên thông tin có trong hệ thống.

Thường quá trình huấn luyện mạng nơron (hay còn được gọi là thuật học) được thực hiện qua phép so sánh đầu ra của mạng với tín hiệu chỉ đạo.

Sau đây là thuật toán học có giám sát (hình 8). Nội dung chính là điều chỉnh trọng số liên kết trong mạng w.



Hình 8: Học có giám sát

Sai số $e = y-d$ là cơ sở để huấn luyện mạng.

2.2 Một số mạng nơron cơ bản

2.2.1 Lớp mạng nơron có giám sát (Supervised neural networks)

2.2.1.1 Perceptron

Mô hình Perceptron do Rosenblatt đưa ra 1958. Mô hình có dạng như đã trình bày trong hình 4. Có một số thuật học cho mạng Perceptron, ví dụ thuật toán cực tiểu bình phương trung bình (LMS-the least mean square), nhưng mô hình còn quá đơn giản nên ít hiệu quả.

2.2.1.2 Perceptron nhiều lớp (the multilayer perceptron–MLP)

Đã có nhiều công trình trình bày nhiều thuật học cho mạng MLP và chỉ rõ rõ sức mạnh của những lớp mô hình này. Cần nhắc tới là thuật toán lan truyền ngược (the Back–Propagation Learning Algorithm) và khả năng xấp xỉ hàm liên tục bằng mạng MLP.

Thuật toán lan truyền ngược cụ thể như sau: Tập dữ liệu đã cho có N mẫu (x_n, d_n), $n = 1, 2, \ldots, N$. Với mỗi n, x_n là tín hiệu đầu vào, d_n là đầu ra mong muốn. Quá trình huấn luyện thực chất là làm cực tiểu hàm G với

$$G = \sum_{n=1}^{N} G_n \,, \quad \text{với} \quad G_n = \frac{1}{N}\sum_{q-1}^{N}(yq(x_n) - dq(x_n))^2 \,.$$

Q là số nút tại lớp ra của mạng. Còn trọng số liên kết mạng được điều chỉnh theo phép lặp sau

$$w(k+1) = w(k) - \mu\frac{\partial G}{\partial w} \,, \qquad \text{trong đó } \mu > 0 \text{ là hằng số tỷ lệ học.}$$

Mạng nơron nhiều lớp lan truyền ngược là một giải pháp hữu hiệu cho công việc mô hình hoá, đặc biệt với quá trình phức tạp hoặc cơ chế chưa rõ ràng. Nó không đòi hỏi phải biết trước dạng hàm hoặc các tham số.

Cũng cần thiết nhắc tới sự kiện sau: Mạng truyền thẳng nhiều lớp được sử dụng để biểu diễn chính xác và biểu diễn xấp xỉ các hàm phi tuyến. Một công cụ cơ bản là định lý sau:

Định lý Kolmogorov

Gọi I_n là khối n chiều của đoạn $[0,1]$, $I_n = [0,1]^n$. Bất kì hàm liên tục $f(x_1, x_2, \cdots, x_n)$ nào của n biến $x_1, x_2, \cdots, x_n$ trên I_n đều có thể biểu diễn dưới dạng

$$f(x_1, x_2, \cdots, x_n) = \sum_{j=1}^{2n-1} h_j \sum_{i=1}^{n} g_{ij}(x_i) \,.$$

ở đây h_j và g_{ij} là hàm liên tục một biến, hơn nữa g_{ij} là hàm tăng đơn điệu cố định không phụ thuộc vào hàm f.

2.2.1.3 Mạng nơron với hàm cơ sở xuyên tâm (mạng RBF)

Hàm cơ sở xuyên tâm (Radial Basis Functions–RBF) đã có từ lâu trong lý thuyết xấp xỉ và được sử dụng để xấp xỉ hàm chưa biết dựa trên cơ sở các cặp điểm vào – ra biểu diễn hàm chưa biết đó.

Trong nhận dạng mô hình hệ thống RBF có thể biểu diễn theo cấu trúc mạng perceptron. Mọi hệ phi tuyến có thể xấp xỉ bằng RBF. Đây là đặc điểm làm cho RBF đặc biệt phù hợp với bài toán nhận dạng mô hình.

Đối với mỗi hàm, việc xấp xỉ được lưu giữ trong các trọng số và tâm của RBF. Tuy nhiên các trọng số này không phải là duy nhất. RBF có biểu diễn toán học như sau :

$$F(x) = C_0 + \sum_{i=0}^{N-1} C_i \varphi(\|x - R_i\|)$$

trong đó

C – véctơ chứa trọng số RBF,

R – véctơ chứa các tâm RBF,

φ – hàm cơ sở hoặc hàm kích hoạt của mạng,

$F(x)$ – hàm nhận được từ đầu ra của mạng,

C_0 – hệ số chệch (có thể là 0),

$\|\cdot\|$ – chuẩn euclidean.

Mỗi tâm R_j có cùng số chiều với véctơ đầu vào x. Các tâm cũng là các điểm bên trong không gian dữ liệu đầu vào và được chọn sao cho chúng là thể hiện của dữ liệu đầu vào. Khi RBF tính toán quá trình xấp xỉ đối với một số điểm dữ liệu đầu vào thì khoảng cách giữa các điểm đầu vào và mỗi tâm được tính theo khoảng cách euclidean. Những khoảng cách này được chuyển qua φ sau đó được trọng số hóa bằng C_i và được tổng hợp lại để sinh ra đầu ra toàn bộ RBF. Một trong những lựa chọn thông thường nhất đối với hàm cơ sở là hàm Gauss:

$\varphi(x) = \exp(-(x-a)^2/2\sigma)$, trong đó σ là tham số tỷ lệ.

Mạng RBF được Moody và Darken đề xuất năm 1989 dựa trên sự tương đồng giữa triển khai hàm cơ sở xuyên tâm và mạng nơron một lớp ẩn.

Nhờ khả năng xấp xỉ các hàm phi tuyến bất kì với độ chính xác tuỳ ý, mạng nơron và sau này là hệ mờ nơron sẽ là công cụ quan trọng, đặc biệt là mạng RBF, cho mô hình hoá hệ thống và cho điều khiển thích nghi các hệ thống phi tuyến.

2.2.1.4 Các mô hình động phi tuyến

Cho đến lúc này ta mới đề cập tới các lớp mạng tĩnh, có giám sát. Song nhiều bài toán thực tiễn đòi hỏi phải xét tới các mô hình động phi tuyến. Nhóm mô hình này có thể chia làm hai lớp: lớp mạng có nối ngược, tức là mạng hồi quy và mạng nơron có thời gian trễ.

1) *Mạng có nối ngược* (mạng hồi quy – mạng Hopfield)

Mạng Hopfield nổi tiếng được bắt đầu nghiên cứu từ 1982. Đây là lớp mạng một lớp với thông tin và quá trình xử lý có nối ngược. Chính công trình của Hopfield đã kích thích làm ra các mạch nơron tích hợp đầu tiên. Mạng Hopfield đã tìm thấy rất nhiều ứng dụng đặc biệt trong bộ nhớ liên hợp và trong các bài toán tối ưu.

Bằng cách xây dựng các thuật toán phức tạp hơn, các mạng hồi quy ngoài các bài toán tối ưu còn tìm thấy nhiều ứng dụng trong dự báo dãy thời gian, nhận dạng các hệ phi tuyến và điều khiển.

Kiến thức cơ sở ban đầu về mạng này xem thêm chương 11 sách [10].

2) *Mạng có thời gian trễ* (mạng TDNN).

Một cách tự nhiên khi xử lý tin chúng ta gặp những tín hiệu xuất hiện theo dãy thời gian. Dĩ nhiên lúc ban đầu người ta dùng mạng nơron tĩnh. Song con đường tất yếu là phải xét tới mô hình động có tham số thời gian. Hơn nữa hệ động không phải là lĩnh vực xa lạ với những người làm về khoa học hệ thống. Do đó mạng có thời gian trễ (mạng TDNN) xuất hiện. Nó đã và sẽ còn được nghiên cứu và ứng dụng trong nhiều bài toán, ví dụ nhận dạng và xử lý tiếng nói.

2.2.1.5 Mạng không giám sát (Unsupervised NN)

Một lớp mô hình nữa của mạng nơron nhân tạo cũng thường được giới thiệu, đó là các mạng nơron có thuật học không có thầy (không giám sát – đôi khi gọi gọn là mạng không giám sát). Như vậy mạng nơron nhân tạo, vì mong muốn bắt chước con người, phải "tự mình" khám phá những mối quan hệ đang quan tâm: những dạng, đường nét, có chuẩn – có bình thường hay không, các hệ số tương quan, ... và sau đó chuyển những quan hệ tìm thấy qua đầu ra. (Để dễ hình dung công việc này chúng ta hãy nhớ tới công việc của các nhà thống kê, nhất là các nhà thống kê dùng máy tính hiện đại).

Như vậy với những mạng này (cũng như nhiều lớp khác), chủ yếu là tìm các thuật học tương ứng với các mạng. Về phần này người ta thường nhắc tới tuật học Hebb, thuật học cạnh tranh, ...

Có thể phân chia số thuật học này thành hai nhóm: nhóm thực hành phân cụm (clustering) và nhóm cố chiết xuất, rút tỉa ra những "đường – nét nào đó" từ dữ liệu.

Trong mạng nơron không giám sát, các thuật học cạnh tranh, mạng tự tổ chức và cả lý thuyết suy diễn thích nghi (Adaptive Resonance Theory – ART) cùng thuộc nhóm một. Các phương pháp chiết xuất theo thành phần chính (Adative Principal Component) thuộc nhóm 2. Bạn đọc tìm thiểu về những lớp mạng này và các thuật học tương ứng hãy xem các sách [10,11,12,16].

2.3 Kết hợp mạng nơron với hệ mờ

Theo dõi sự phát triển trong thời kỳ còn phát triển độc lập người ta dễ nhận thấy cả hai lý thuyết và do đó cả hai công nghệ đều có những mục đích gần gũi, hơn nữa những thành đạt cũng có những điểm tương tự. Do vậy sự kết hợp với nhau là lẽ rất tự nhiên.

Sau đây là mấy điểm tương đồng giữa hệ trên cơ sở logic mờ và mạng nơron nhân tạo:

– Cả hai đều nhằm tăng thêm tri thức, tăng độ thông minh cho các hệ thống với sự giúp đỡ của các hệ thống kỹ thuật (đặc biệt với máy tính), trong môi trường bất định, có nhiễu, thông tin và tri thức thiếu chính xác.

- Cả hai đều là hệ động, là công cụ ước lượng bằng số không dùng mô hình số chọn trước.
- Cả hai khi ước lượng hàm số không đòi hỏi mô tả dạng toán học $y = f(x)$, thường " học được " từ mẫu dữ liệu, tiệm cận với các số liệu, khác nhiều với cách tiếp cận xử lý tín hiệu như trong phần Trí tuệ nhân tạo (AI) ở giai đoạn trước.
- Cả hai loại hệ thống và công nghệ đều rất thành đạt, đã đưa ra nhiều hệ thống và thiết bị đang dùng trong đời sống hàng ngày.

Sự kết hợp các hệ mờ với mạng nơron ,ít nhất cũng đề xuất ngay được một ý tưởng mới: chuyển nhiệm vụ thiết kế nhiều phần của hệ mờ trong các bộ điều khiển và hệ trợ giúp quyết định thành các bài toán huấn luyện và học trong các mạng nơron. Như vậy sẽ gặt hái được tất cả ưu thế của cả hai lý thuyết và hai công nghệ.

Sau đây là phác họa gọn mấy ý về kết hợp hai công nghệ.

2.3.1 Mạng nơron mờ

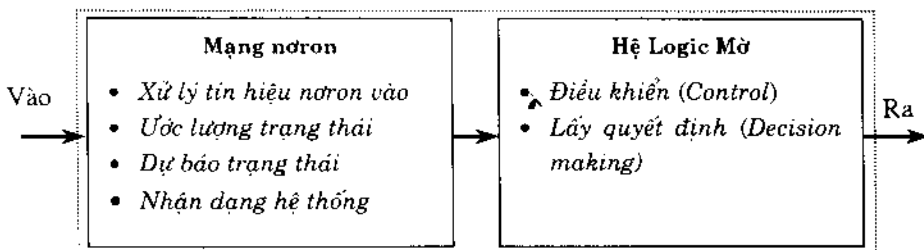
Sự kết hợp trực quan đầu tiên là trực tiếp suy rộng mạng nơron bằng cách đưa các khái niệm mờ đặc biệt là tập mờ và số mờ vào mạng nơron và xem xét xem những bài toán nào, thuật toán nào còn đúng. Tác động của lớp thuật toán mới ra sao ?

Hoàn toàn tự nhiên người ta nghĩ ngay tới và nghiên cứu bốn loại suy rộng sau:

1) *Loại 1*: Tín hiệu vào là số thực, trọng số mờ.
2) *Loại 2*: Ttín hiệu vào là tập mờ, trọng số là số thực.
3) *Loại 3*: Cả tín hiệu vào và trọng số đều là mờ.
4) *Loại mở rộng*: Khai thác các phép toán t–chuẩn, t–đối chuẩn.

2.3.2 Hệ mờ nơron

Cần kết hợp nhuần nhuyễn hơn hệ mờ với mạng nơron nhân tạo. Kết quả phụ thuộc vào từng nghiên cứu,vào từng kiến trúc của thiết kế lớp hệ thông mới. Hình 9 là một kiến trúc kiểu mẫu.



Hình 9: Kiến trúc kiểu mẫu của một hệ mờ nơron.

Không còn nghi ngờ gì điều khẳng định sau: "trong lĩnh vực các hệ thống mờ nơron còn nhiều khoảng trống mênh mông đầy sức hấp dẫn cho sức sáng tạo của những ai quan tâm nghiên cứu, thiết kế, cài đặt và ứng dụng các hệ tri thức".

Bạn đọc có thể tìm thấy những kiến thức ban đầu về các hệ mờ nơron trong các tài liệu [9,10,17].

Tài liệu trích dẫn cho Phần 1

[1] **L.A. Zadeh**: Fuzzy sets, Inform. and Control ,8, 1965, 338-353.

[2] **D. Dubois and H. Prade**: Fuzzy Sets and Systems, Academic Press, N.Y. , 1980.

[3] **H.J. Zimmermann**: Fuzzy Set Theory - and Its Applications, 2 nd Ed., Kluwer Acad. Pub., Dodrech, 1991.

[4] **Hệ mờ và ứng dụng**: Biên tập tập thể : Nguyễn Hoàng Phương, Bùi Công Cường, Nguyễn Doãn Phước, Phan Xuân Minh và Chu Văn Hỷ. NXB Khoa học và Kỹ thuật, 1998, Hà nội.

[5] **Anca L. Ralescu, Ed.**: Applied research in fuzzy technology. Kluwer Acad. Pub.. Dordrecht, 1994.

[6] **Satoru Fukami and Minoru Yoneda**: Decision Support System, [5] trang 17-66.

[7] **Bùi công Cường**: Cơ sở toán học của các hệ mờ.Giáo trình. ĐH Bách khoa Hà nội, 1998-1999.

[8] **Bùi công Cường**: Một số kiến thức cơ sở của logic mờ trong các hệ mờ.trong [4], trang 1-21.

Tài liệu trích dẫn cho Phần 2

[1] **B.Kosko**: Neural Networks and Fuzzy Systems,Prentice-Hall, NJ,1992.

[2] **C.T. Lin and C. S. G. Lee**: Neural Fuzzy systems, Prentice Hall. London,1996.

[3] **R. Schalkoff**: Artifical Neural Networks, McGrawhill, Singapore,1997.

[4] **V.Roychodhury, K.Y.Siu, A.Orlitsky**: Theoretical Advancess in Neural Computation and Learning, Kluwer Acad.Pub., Boston, 1994.

[5] **Gyorgrry Turan**: [12], trang 243 ÷ 294.

[6] **Wolfgang Maass**: [12]. 295 ÷ 336.

[7] **Vũ Như Lân**: Một số vấn đề nhận dạng mô hình và điều khiển sử dụng mạng nơron. Trường Thu – Hệ mờ và ứng dụng, Hà nội , 8/2000.

[8] **N.K.Bose and P.Liang**: Neural Network Fundamentals with Graph.Algorithms,and Applications, McGrawhill, New York,1996.

[9] **C.H.Chen (Ed.)**: Fuzzy Logic and Neural Network Handbook,McGraw-Hill, NewYork, 1996.

2 LÝ THUYẾT TẬP MỜ VÀ CÔNG NGHỆ TÍNH TOÁN MỀM

Nguyễn Cát Hồ
Viện Công nghệ Thông tin

Cách mạng KHKT về cơ khí ra đời đã đem đến năng suất lao động mới và sự phát triển kinh tế–xã hội có tính cách mạng. Ngày nay chúng ta vẫn tiếp tục chứng kiến những thành tựu nghiên cứu phát triển các công cụ, thiết bị với công nghệ hiện đại, đặc biệt các thiết bị và dây truyền sản xuất tự động hoá nhằm tăng năng suất và thay thế sức lao động của con người. Có thể xem cách mạng cơ khí và tự động hoá như là biện pháp 'kéo dài bàn tay' của con người để tăng năng suất lao động.

Theo logic lý tự nhiên, sự phát triển khoa học và kỹ thuật lại dẫn đến khả năng 'kéo dài' năng lực tư duy, suy luận của con người. Bằng năng lực tư duy của mình con người đã và đang khai phá thế giới thực tế rộng lớn. Thế giới hiện thực và tri thức khoa học cần khám phá là vô hạn và là những hệ thống cực kỳ phức tạp, nhưng ngôn ngữ mà năng lực tư duy và tri thức của chúng ta sử dụng làm phương tiện nhận thức và biểu đạt lại chỉ hữu hạn. Lịch sử phát triển sáng tạo của loài người chỉ ra rằng phương tiện ngôn ngữ tuy hữu hạn nhưng đủ để cho con người mô tả, nhận thức các sự vật, hiện tượng để tồn tại và phát triển. Như là một hệ quả tất yếu của việc sử dụng một số lượng hữu hạn các từ ngữ của một ngôn ngữ tự nhiên để mô tả tính vô hạn các sự vật hiện tượng, dễ nhận thấy rằng hầu hết các bài toán liên quan đến hoạt động nhận thức, trí tuệ của con người đều hàm chứa những đại lượng, thông tin mà bản chất là không chính xác, không chắc chắn, không đầy đủ. Sẽ chẳng bao giờ có các thông tin, dữ liệu cũng như các mô hình toán – lý đầy đủ và chính xác cho các bài toán dự báo thời tiết. Và nhìn chung con người luôn ở trong bối cảnh thực tế là không thể có thông tin đầy đủ và chính xác cho các hoạt động lấy quyết định của mình và cũng không thể hy vọng có những quyết định đúng đắn và chính xác như các mệnh đề, định luận trong khoa học toán–lý hay nói chung khoa học tự nhiên.

Như vậy có thể thấy có rất nhiều vấn đề rộng lớn trong thực tiễn, liên quan đến hầu hết các lĩnh vực khoa học kỹ thuật, nhiều hay ít đều hàm chứa những yếu tố có bản chất không đầy đủ, không chắc chắn.

Mỗi lĩnh vực khoa học kỹ thuật đều có một miền ứng dụng của mình. Khoa học kỹ thuật lấy tính "chính xác" làm cơ sở xây dựng và phát triển sẽ có một miền ứng dụng và cũng có những giới hạn xác định không thể vượt qua và nó chỉ có khả năng mô phỏng được một phần thế giới thực tế. Liệu có một lý thuyết toán học nào cho phép mô hình hoá phần thế giới thực mà con người vẫn chỉ có thể nhận thức, mô tả bằng ngôn ngữ tự nhiên vốn hàm chứa những thông tin không chính xác, không chắc chắn hay không?

Phát hiện thấy nhu cầu tất yếu ấy, năm 1965 L.A. Zadeh đã sáng tạo ra lý thuyết tập mờ (*Fuzzy Sets Theory*) và đặt nền móng cho việc xây dựng một loạt các lý thuyết quan trọng

dựa trên cơ sở lý thuyết tập mờ. Kể từ đấy một trào lưu khoa học lấy tính không chắc chắn, không chính xác làm triết lý để nghiên cứu sáng tạo đã phát triển mạnh mẽ, và người ta đánh giá rằng những công trình của Zadeh như là một trong những phát minh quan trọng có tính chất bùng nổ và đang hứa hẹn giải quyết được nhiều vấn đề phức tạp và to lớn của thực tiễn. Như một nhà khoa học hệ thống tổng quát Mỹ George Klir đã nhận định *chỉ cần làm chủ một chút tính không chắc chắn cũng có thể giải quyết được những vấn đề rất to lớn*.

Một trong những vấn đề như vậy là việc mô hình hoá các qúa trình tư duy, lập luận của con người để tự động hoá hỗ trợ cho các hoạt động tư duy, chẳng hạn như hoạt động lấy quyết định. Nhiều hệ chuyên gia hay hệ trợ giúp quyết định đã được phát triển và đưa vào ứng dụng. Nhiều thiết bị thông minh được thiết kế xây dựng dựa trên công nghệ tập mờ, logic mờ đã xuất hiện trên thị trường và được ứng dụng trong lĩnh vực chế tạo xe ô tô, các thiết bị tiêu dùng, trong điều khiển tự động trong các nhà máy .... Sự phát triển mạnh mẽ các ứng dụng đã dẫn đến việc thành lập ở nhiều nước các phòng thí nghiệm để phát triển các ứng dụng công nghệ mờ trong công nghiệp. Đáng chú ý là phòng thí nghiệm LIFE (Laboratory for International Fuzzy Engineering) ở Nhật mà cái tên viết tắt của nó thể hiện một niềm tin rằng công nghệ này là công nghệ của *Cuộc sống* trong tương lai. Ngày nay không chỉ các nước phát triển mà ngay cả những nước đang phát triển cũng quan tâm nghiên cứu và phát triển các ứng dụng của các "lĩnh vực khoa học mờ" như Trung quốc, Singapor, Brazil, Ai cập, Iran .... Điều này chứng minh thêm ý nghĩa thực tiễn của lĩnh vực "khoa học mờ".

Tuy mục tiêu nguyên thủy của việc ra đời lý thuyết tập mờ là ứng dụng tự động hoá các hoạt động tư duy của con người, nhưng về mặt lý thuyết nó lại là một sự mở rộng rất chỉnh, rất đẹp đẽ của khái niệm tập hợp kinh điển. Như chúng ta đã biết, lý thuyết tập hợp kinh điển là cơ sở, nền tảng cho việc hình thức hoá một cách nhất quán và cho sự phát triển của các ngành toán học và do đó cho các ngành khoa học khác. Như là một hệ quả logic, hầu như tất cả các ngành khoa học này có người em sinh đôi được mở rộng và phát triển trên cơ sở lý thuyết tập mờ. Nêu một vài ví dụ như giải tích mờ, lý thuyết các hệ vi tích phân mờ, tôpô mờ, lý thuyết nhóm mờ, lý thuyết điều khiển mờ, ...

Với vai trò và khả năng to lớn của việc phát triển các ứng dụng đa dạng của lý thuyết tập mờ, và vì đây là bài tổng quan đầu tiên mà chúng tôi giới thiệu với bạn đọc thuộc các lĩnh vực khoa học khác nhau và với sự hiểu biết có hạn, chúng tôi chỉ xin trình bày những nét tổng quan về lý thuyết tập mờ và một số ngành khoa học được xây dựng và phát triển trên nền tảng của lý thuyết tập mờ với hy vọng đem lại sự quan tâm của bạn đọc đối với lĩnh vực khoa học vẫn nên xem là còn mới mẻ ở nước ta.

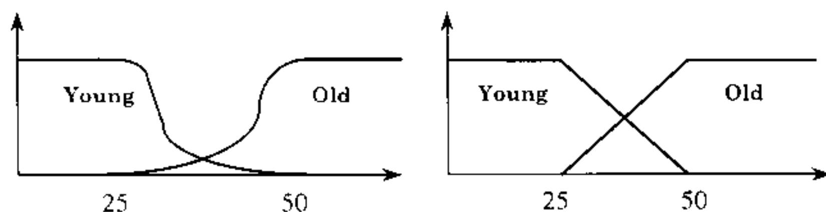
Để đơn giản việc trình bày chúng tôi sẽ bỏ qua các chi tiết quá hình thức hoá của tính chính xác, tính chặt chẽ làm người đọc khó nắm bắt các ý tưởng chính. Bạn đọc nào muốn quan tâm nghiên cứu sâu hơn, cần nội dung chính xác đầy đủ hơn có thể tìm tài liệu trong phần phụ lục các tài liệu được xắp xếp theo chuyên ngành.

1 Lý thuyết tập mờ và logic mờ là cơ sở phương pháp luận cho việc giải các bài toán với các đại lượng không chính xác (inexact), không chắc chắn (uncertain).

1.1 Tập mờ và ngữ nghĩa khái niệm mờ

Như trên đã nói. con người sử dụng ngôn ngữ có số lượng ngữ nghĩa (do số lượng các từ) hữu hạn để nhận biết, nhận thức. phản ảnh thế giới vô hạn. Về triết lý, điều này dẫn đến ngôn ngữ về bản chất vốn chứa những thông tin không chính xác, không chắc chắn. Về nhận thức khoa học, điều này khẳng định có khả năng thực tế sử dụng ngôn ngữ để mô hình hoá thế giới thực trong những lĩnh vực lý thuyết nào đó.

L.A. Zadeh đã sáng lập lý thuyết tập mờ, trong đó tập mờ được xem như là một sự mở rộng trực tiếp của tập hợp kinh điển: Tập hợp kinh điển $A \subseteq U$ có một danh giới sắc nét, rõ ràng. và vì vậy nó được biểu thị bằng hàm đặc trưng $\mu_A(u) = \begin{cases} 1 \text{ if } u \in A \\ 0 \text{ if } u \notin A \end{cases}$. Ví dụ, A là tập những người có tuổi dưới 19 (điều kiện cần để tham gia đội bóng U19!) là một tập hợp kinh điển. Mỗi người (phần tử) chỉ có hai khả năng rõ ràng: hoặc là phần tử của A hoặc không. Tuy nhiên ta xét tập $\tilde{A}$ gồm những người là trẻ. Trong trường hợp này sẽ không có ranh giới rõ ràng để khẳng định một người có là phần tử của $\tilde{A}$ hay không: Ranh giới của nó là mờ. Ta chỉ có thể nói một người sẽ thuộc tập hợp $\tilde{A}$ ở một mức độ nào đó. Chẳng hạn chúng ta có thể đồng ý với nhau (subjective) rằng một người 35 tuổi thuộc về tập hợp $\tilde{A}$ với độ thuộc là 60% hay 0,6. Và Zadeh gọi một tập $\tilde{A}$ (hay có thể kí hiệu là $\underline{A}$ để phân biệt với tập kinh điển A) như vậy *là tập mờ* và đồng nhất tập hợp $\tilde{A}$ với một hàm $\mu_{trẻ}$: $Y \to [0,1]$, gọi là hàm thuộc của tập $\tilde{A}$ (*memberhip function*), trong đó Y là tập số tự nhiên dùng để đo độ tuổi tính theo năm. gọi là không gian tham chiếu. Từ *trẻ* gọi là khái niệm mờ (vague concept). Như vậy mọi phần tử đều thuộc vào tập *trẻ* ở một mức độ nào đó!



Hình 1: Khái niệm tập mờ.

Về mặt ngữ nghĩa, hàm thuộc cho ta khả năng biểu thị trực cảm (intuition) của chúng ta về ý nghĩa của khái niệm mờ. Nhưng tại sao một khái niệm mờ lại được biểu thị bằng một hàm thuộc này mà không phải là một hàm khác. Có thể thấy không thể xác định chính xác được hàm thuộc cho một khái niệm mờ. Vì vậy người ta nói hàm thuộc có tính chất chủ

quan (subjective) và Zadeh đưa ra ý tưởng là việc chấp nhận một khái niệm mờ được biểu thị bằng một tập mờ (hàm thuộc) là một ràng buộc (constraint).

Nhìn chung phần lớn các khái niệm mờ đều có thể dễ dàng biểu thị bằng những tập mờ và do đó lý thuyết tập mờ cho ta tiềm năng mô hình hoá toán học một lớp rộng lớn các bài toán thực tế, đặc biệt các bài toán liên quan đến tư duy, đến quá trình lấy quyết định của con người.

Trong bất kỳ ngôn ngữ của một dân tộc nào cũng có các khái niệm mờ mà là các trạng từ, có vai trò biến đổi ngữ nghĩa của các từ khác ở một mức độ nào đó như *very, slightly, more or less*, ... (gọi là các gia tử), hay các liên từ logic như *and, or, if ... then*, ... Chúng ta có thể định nghĩa các phép tính trên tập mờ (t.l. trên các hàm) để biểu thị ngữ nghĩa của các gia tử, các phép liên kết logic ... như xuất hiện trong các biểu thức $\underline{A}$ AND $\underline{B}$, $\underline{A}$ OR $\underline{B}$, IF $\underline{A}$ THEN $\underline{B}$, NOT $\underline{A}$, *VERY* $\underline{B}$, $\underline{A}$ *Approx.* $\underline{A}$, *More–or–Less* $\underline{B}$, ..., *About* 30 year old, *Almost young* people are *intelligent* (các chữ viết nghiêng là các khái niệm mờ (vague)). Về nguyên tắc việc chọn các phép tính để biểu thị ngữ nghĩa của các phép liên kết logic trên là khá tự do, tuỳ theo trực cảm (intuition) của nhóm người sử dụng (ví dụ cảm nhận của những người lái tầu hoả đối với một loại tàu nhất định về khoảng cách đến đích hay vật chướng ngại, về tốc độ của con tàu, về tính năng của con tàu và các cơ chế điều khiển như kỹ năng sử dụng phanh tàu), để đạt được ý nghĩa mà họ mong muốn. Trong việc thiết kế các hệ điều khiển mờ việc này còn có thể đạt được bằng việc nghiên cứu thử nghiệm.

1.2 Đại số các tập mờ

Gọi $F(U,[0,1])$ là tập tất cả các tập mờ trên U. Có thể hiểu việc biểu diễn ngữ nghĩa các từ ngôn ngữ bằng tập mờ như là một ánh xạ định lượng ngữ nghĩa từ miền ngôn ngữ vào không gian hàm $F(U,[0,1])$. Đây là một không gian rất giàu về cấu trúc tính toán và vì vậy thủ tục xử lý thông tin ngôn ngữ và tư duy lập luận của con người – mà nó có vẻ còn phức tạp, bí hiểm vì chúng ta còn chưa khám phá được gì nhiều về nó – sẽ được mô phỏng bằng các thủ tục dựa trên cấu trúc tính toán của $F(U,[0,1])$. Để linh hoạt và mềm dẻo trong việc mô phỏng các quá trình phức tạp như vậy sẽ có tính khá tự do trong các định nghĩa các phép tính trên tập mờ.

1.2.1 Các phép tính

Chúng tôi xin giới thiệu ở đây một số phép tính trên các tập mờ:

	Phép toán trên tập mờ	**Định nghĩa trên hàm thuộc**
1)	$\underline{A} \subseteq \underline{B}$	$\mu_A(u) \le \mu_B(u)$
2)	$\underline{A} \cup \underline{B}$	$\mu_{A \cup B}(u) = \mu_A(u) \vee \mu_B(u) = \max\{\mu_A(u), \mu_B(u)\}$
		$\mu_{A \cup B}(u) = \mu_A(u) + \mu_B(u) - \mu_A(u)\mu_B(u)$
3)	$\underline{A} \cap \underline{B}$	$\mu_{A \cap B}(u) = \mu_A(u) \wedge \mu_B(u) = \min\{\mu_A(u)\ ,\ \mu_B(u)\ \}$

$$\mu_{A \cap B}(u) = \mu_A(u)\mu_B(u)$$

4) $-\underline{A}$ (hay $\complement A$, phần bù của $\underline{A}$) $\mu_{-A}(u) = 1 - \mu_A(u)$

5) $\underline{A} \oplus \underline{B}$ $\mu_{A \oplus B}(u) = \mu_A(u) + \mu_B(v) - \mu_A(u)\mu_B(v)$

6) U $\mu_U(u) \equiv 1$

7) $\varnothing$ $\mu_\varnothing(u) \equiv 0$

8) Tích Đe-cac $\underline{A} \times \underline{B}$ $\mu_{A \times B}(u,v) = \min\{\mu_A(u)\ ,\ \mu_B(v)\}$

1.2.2 Các phép t–norm *t* và t–conorm *s*

Theo xu thế nói ở trên, để giảm bớt sự phụ thuộc vào các phép tính min và max và do đó làm tăng tính mềm dẻo và linh hoạt trong việc giải các bài toán thực tế người ta mở rộng hai phép tính min, max thành hai lớp hàm t–norm và t–conorm có từng cặp phần tử đối ngẫu nhau.

– Chẳng hạn bỏ qua đặc tính cụ thể trong việc xác định giá trị của hàm min hai biến mà chỉ giữ lại một số tính chất của nó ta thu được lớp hàm t–norm $t(x,y)$:

a) Giao hoán: $t(x,y) = t(y,x)$

b) Kết hợp: $t(x,t(x,z)) = t(t(x,y),z)$

c) Tồn tại phần tử đơn vị: $t(1,y) = y$

d) Không giảm theo từng biến.

– Một cách tương tự và đối ngẫu, bỏ qua tính cụ thể trong việc xác định giá trị của hàm Max hai biến mà chỉ giữ lại một số tính chất của nó ta thu được lớp hàm t–conorm $s(x,y)$:

a) Giao hoán: $s(x,y) = s(y,x)$

b) Kết hợp: $s(x,s(x,z)) = s(s(x,y),z)$

c) Có phần tử không: $s(0,y) = y$

d) Không giảm theo từng biến.

Có thể thấy t–norm và t–conorm tuân thủ quy tắc đối ngẫu nhau theo nghĩa

$$t(x,y) = 1 - s(1-x, 1-y)$$

Như vậy chúng ta sẽ không có một đại số tập mờ duy nhất, vì trong định nghĩa đại số tập hợp ta luôn có thể thay min, max bằng t–norm và t–conorm đối ngẫu nhau và thu được một đại số tập mờ khác.

1.3 Quan hệ mờ

Quan hệ mờ n ngôi là một tập mờ R trong không gian tích Đe-cac của n không gian $U_1 \times U_2 \times \cdots \times U_n$.

Quan hệ mờ 2 ngôi $R(u,v)$ gọi là

a) *Đối xứng* nếu $\mu_R(u,v) = \mu_R(v,u)$.

b) *Phản xạ* nếu $\mu_R(u,u) = 1, \forall u \in U$.

c) *Phản phản xạ* nếu $\mu_R(u,u) = 0, \forall u \in U$.

d) *Bắc cầu Max–Min* nếu $\mu_R(u,v) \geq \vee\{\mu_R(u,w) \wedge \mu_R(w,v) : w \in U\}$.

e) *Bắc cầu Min–Max* nếu $\mu_R(u,v) \leq \wedge\{\mu_R(u,w) \vee \mu_R(w,v) : w \in U\}$.

f) *Phép hợp thành RoS* với $R \in U \times W$, $S \in W \times U$ được định nghĩa như sau

$$\mu_{RoS}(u,v) = \vee\{\mu_R(u,w) \wedge \mu_S(w,v) : w \in U\}.$$

R là quan hệ *tương tự* (quan hệ tương đương mờ) nếu nó có các tính chất:

- Phản xạ.
- Đối xứng.
- Bắc cầu.

R là quan hệ *không tương tự* nếu nó là phần bù của một quan hệ tương tự hay, một cách tương đương, nó thoả mãn các tính chất:

- Phản phản xạ.
- Phản đối xứng.
- Bắc cầu Min–Max.

Ký hiệu $d_R(u,v) = \mu_R(u,v)$. Khi đó chúng ta thấy nó có các tính chất sau:

a) $d_R(u,v) \geq 0$.

b) $d_R(u,v) = d_R(v,u)$.

c) $d_R(u,v) = d_R(u,w) \vee d_R(w,v)$.

Như vậy $d_R(u,v)$ có các tính chất tương tự như khoảng cách trong không gian hình học và được gọi là khoảng cách Min–Max.

R là quan hệ *giống nhau* nếu nó thỏa mãn:

- Phản xạ.
- Đối xứng.

R là quan hệ *không giống nhau* nếu nó là phần bù của quan hệ giống nhau, tức là:

- Phản phản xạ.
- Đối xứng.

Như vậy bao đóng bắc cầu $R^\wedge$ của quan hệ giống nhau là quan hệ tương tự. $R^\wedge$ được tính theo công thức:

$$R^{\wedge} = R^1 \cup R^2 \cup \cdots R^k \cup \cdots$$

trong đó:

$$R^2 = RoR \quad \text{với} \quad \mu_{RoS}(u,v) = \vee\{\mu_R(u,w) \wedge \mu_R(w,v) : w \in U\}.$$

$$R^k = R^{k-1} oR.$$

Định lý 1: (i) Nếu có n mà $R^n = R^{n+1}$ thì $R^{\wedge} = R^1 \cup R^2 \cup \cdots R^n$.

(ii) Nếu U hữu hạn và $/U/ = n$ thì $R^{\wedge} = R^1 \cup R^2 \cup \cdots R^n$.

Định lý 2: Nếu R là quan hệ mờ tương tự thì tập mức α, $R_\alpha = \{u, v \in U: \mu_R(u,v) \geq \alpha\}$ của R là quan hệ tương đương kinh điển và do đó nó phân hoạch U thành các lớp tương đương.

Ký hiệu R' là phần bù của R, tức là $R' = \mathcal{C}R$ thì R' là quan hệ không tương tự và

$$R'_{1-\alpha} = \{u, v \in U: \mu_{R'}(u,v) = d_R(u,v) \leq 1-\alpha\} = R_\alpha$$

nói cách khác, khoảng cách giữa hai phần tử bất kỳ trong cùng lớp tương đương không vượt quá $1-\alpha$.

Phân hoạch các đối tượng mờ:

Với cơ sở lý thuyết trên ta đã có thể có bài toán ứng dụng sau. Xét các đối tượng mờ A_i, $i=1,2, \ldots, n$, nghĩa là các đối tượng được biểu thị bằng tập mờ. Chẳng hạn các đối tượng đó là học sinh với điểm các môn học là (ở đây điểm $x/10$ hiểu là $0.x$):

S/V	V=văn	T=toán	Đ=địa	S=sử	L= lý	S=sinh	H= hoá	N=n/ngữ
A_1	0.5	0.8	0.7	0.6	0.9	0.6	0.9	0.9
A_2	1.0	0.6	0.8	0.7	0.6	0.8	0.6	0.8
A_3	0.8	0.7	0.9	0.8	0.6	0.9	0.8	0.7
A_4	0.7	0.9	0.6	0.7	0.8	0.9	0.6	0.6
A_5	0.8	0.6	0.9	0.8	0.7	0.7	0.7	0.8

Như vậy mỗi học sinh được xem như là một đối tượng mờ. Mục đích của bài toán là phân loại các đối tượng có "độ tương tự nhau" vào cùng một lớp phân hoạch theo một tiêu chuẩn nào đó. Về ý nghĩa thực tiễn ta mong muốn các học sinh trong cùng một nhóm sẽ có năng khiếu tương tự nhau.

Trên cơ sở phương pháp luận trên, việc phân loại sẽ được tiến hành như sau:

– Xây dựng *quan hệ không tương tự* giữa các đối tượng mờ:

a) Định nghĩa khoảng cách giữa các đối tượng: Giả sử A và B là hai đối tượng mờ bất kỳ. Ta định nghĩa khoảng cách giữa A và B là khoảng cách Hamming

$$d(A,B) = \frac{1}{n}\left(|A(u_1) - B(u_1)| + |A(u_2) - B(u_2)| + \ \dots + |A(u_n) - B(u_n)|\right),\ 0 \le d(A,B) \le 1.$$

b) Ký hiệu $d_{ij} = d(A_i, A_j)$ và lập *quan hệ không giống nhau*

$R = (d_{ij})_{n\times m}$.

c) Tính $R^* = \mathcal{C}((\mathcal{C}R)^\wedge)$. Đây là quan hệ không tương tự mà mỗi phần tử của ma trận chính là khoảng cách giữa các đối tượng mờ.

– Ta thấy $R^*_\alpha = \{i,j\ :\ d_{R^*}(i,j) \le \alpha\}$ là một quan hệ tương đương kinh điển và nó xác định một phân hoạch mức α trên các đối tượng. Như vậy các đối tượng trong cùng một lớp phân hoạch có khoảng cách (mức độ không tương tự $\le \alpha$, hay mức độ tương tự $\ge (1-\alpha)$ không vượt quá α.

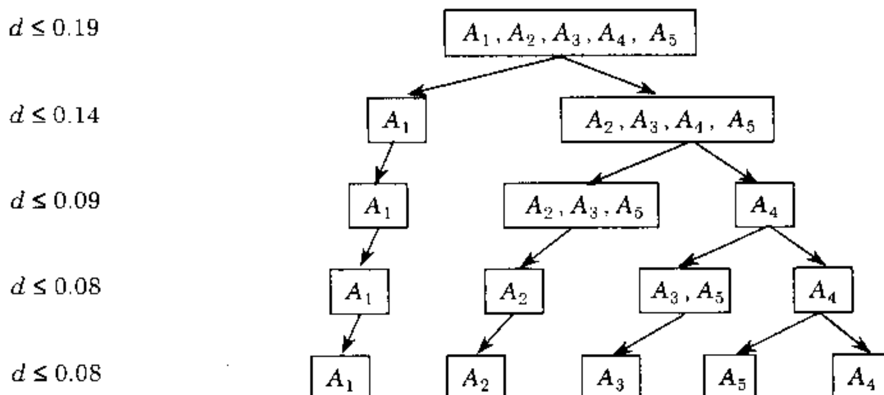
– Xây dựng cây phân tích.

Trong ví dụ trên ta tính được các kết quả sau:

$$8R = 8(d_{ij})_{n\times m} = \begin{vmatrix} 0.0 & 1.8 & 1.7 & 1.5 & 1.5 \\ 1.8 & 0.0 & 0.9 & 1.3 & 0.7 \\ 1.7 & 0.9 & 0.0 & 1.1 & 0.6 \\ 1.5 & 1.3 & 1.1 & 0.0 & 1.4 \\ 1.5 & 0.7 & 0.6 & 1.1 & 0.0 \end{vmatrix},\quad 8(\mathcal{C}R)^2 = 8(\mathcal{C}R)^3 = \begin{vmatrix} 8.0 & 6.5 & 6.5 & 6.5 & 6.5 \\ 6.5 & 8.0 & 7.3 & 6.9 & 7.3 \\ 6.5 & 7.3 & 8.0 & 6.9 & 7.4 \\ 6.5 & 6.9 & 6.9 & 8.0 & 6.9 \\ 6.5 & 7.3 & 7.4 & 6.9 & 8.0 \end{vmatrix}$$

$$R^* = \begin{matrix} 0.00 & 0.19 & 0.19 & 0.19 & 0.19 \\ 0.19 & 0.00 & 0.09 & 0.14 & 0.09 \\ 0.19 & 0.09 & 0.00 & 0.14 & 0.08 \\ 0.19 & 0.14 & 0.14 & 0.00 & 0.14 \\ 0.19 & 0.09 & 0.08 & 0.14 & 0.00 \end{matrix} \qquad \begin{matrix} d \le 0.19 & \{A_1, A_2, A_3, A_4, A_5\} \\ d \le 0.14 & \{A_1\}, \{A_2, A_3, A_4, A_5\} \\ d \le 0.09 & \{A_1\}, \{A_2, A_3, A_5\}, \{A_4\} \\ d \le 0.08 & \{A_1\}, \{A_2\}, \{A_3, A_5\}, \{A_4\} \\ d \le 0.08 & \{A_1\}, \{A_2\}, \{A_3\}, \{A_5\}, \{A_4\} \end{matrix}$$

Như vậy ta thu được cây phân tích sau:



Với những nội dung đã trình bày trong phần này chúng ta có thể đưa ra một số nhận xé
sau:

- Về mặt toán học lý thuyết tập mờ là một sự khái quát trực tiếp của lý thuyết tập hợ kinh điển.
- Nó không phải là một lý thuyết duy nhất, và vì vậy nó mô tả một lớp các phép toán một lớp các đại số tập mờ, ...
- Nó tạo ra khả năng mô tả định lượng ngữ nghĩa của ngôn ngữ và vì vậy nó cho phé mô phỏng những bài toán không có hoặc khó khăn mới có mô hình định lượng, chẳn hạn các bài toán được mô tả bằng ngôn ngữ. Đây chính là đặc điểm quan trọng ch phép phương pháp luận tập mờ tiếp cận mềm dẻo các bài toán thực tiễn phức tạp, đ dạng trong hầu hết các lĩnh vực của cuộc sống.

2 Toán học mờ (Fuzzy Mathematics)

Như trên chúng tôi đã trình bày, khái niệm tập mờ về hình thức hoá là một sự khái qu rất chỉnh và vì vậy những lý thuyết toán học xây dựng trên cơ sở tập hợp thông thường đề có thể chuyển sang lý thuyết mờ tương ứng được xây dựng trên lý thuyết các tập mờ. Tu nhiên như chúng tôi đã trình bày ở trên, chúng ta không có một lý thuyết tập mờ duy nh và vì vậy nói chung chúng ta luôn luôn có nhiều hướng mở rộng các lý thuyết kinh điển nh dưới đây chúng ta sẽ thấy.

2.1 Topo mờ

Một trong những lý thuyết cơ sở quan trọng để phát triển toán học là lý thuyết cá không gian topo. Một yêu cầu tự nhiên theo trào lưu này là cần xây dựng và phát triển thuyết các không gian topo mờ (xem Mingsheng Ying: A new approach for fuzzy topolog Vol. 55, N2(1993); W. Gahler, A.S. Abd-Allah and A. Kandel: On extended fuzz topologies; V. Gregori and A. Vidal: Gradations of openness and Chang fuzzy topologie Vol.109, N2(2000)).

Giả sử X là một tập hợp bất kỳ và $I = [0,1]$ là đoạn thẳng đơn vị. Ta ký hiệu I^X là tập t cả các tập mờ của X.

Định nghĩa 1: (Topo mờ "kinh điển") Một họ các tập mờ $T \subsetneq I^X$ được gọi là topo mờ nếu thoả các tiên đề sau:

a) $\varnothing, X \in T$.

b) Nếu $\underline{A}, \underline{B} \in T$ thì $\underline{A} \cap \underline{B} \in T$.

c) Nếu $\underline{A}_i \in T$ với $\forall i \in \xi$ thì $\bigcup_{i \in \xi} \underline{A}_i \in T$.

Cặp (X, T) gọi là không gian topo theo nghĩa "kinh điển".

Một ánh xạ $C\ell : I^X \to I^X$, được gọi là một toán tử lấy bao đóng nếu nó thoả mãn:

a) $C\ell(\varnothing) = \varnothing$.

b) $C\ell(C\ell(\underline{A})) = C\ell(\underline{A})$.

c) $C\ell(\underline{A}) \supseteq \underline{A}$.

d) $C\ell(\underline{A} \cup \underline{B}) = \underline{A} \cup C\ell(\underline{B})$.

Tương tự như trong trường hợp topo kinh điển. toán tử $C\ell$ sẽ cảm sinh một topo

$$T_{C\ell} = \{\underline{A} \in I^X : \underline{A} = \mathcal{C}(C\ell(\mathcal{C}\underline{A}))\}.$$

Lân cận của một điểm mờ: Tập mờ $x_t \in I^X$ với $x \in X$, $\mu_{x_t}(x) = t$, $\mu_{x_t}(y) = 0$, $x \neq y$, được gọi là một điểm mờ trong I^X. Giống như quan hệ bao hàm, ta nói điểm mờ $x_t \in \underline{A}$ nếu $x_t \subseteq \underline{A}$, nghĩa là khi $t \leq \mu_A(x)$. Một tập mờ $\underline{A} \in I^X$ được gọi là một lân cận của điểm mờ x_t nếu $\exists U \in T$ sao cho $x_t \in U \subseteq \underline{A}$.

Định nghĩa 2: (Topo phân bậc tính mở) Ánh xạ $\tau : I^X \to I (I = [0,1])$ được gọi là sự phân bậc tính mở (gradation of openess) nếu nó thoả mãn các điều kiện sau:

a) $\tau(\varnothing) = \tau(X) = 1$,

b) $\tau(\underline{A} \cap \underline{B}) \geq \tau(\underline{A}) \cap \tau(\underline{B})$,

c) $\tau\left(\bigcup_i A_i\right) \geq \bigcap_i \tau(A_i)$.

Khi đó (X, τ) gọi là không gian topo mờ phân bậc.

Định lý 3: Giả sử (X, τ) là không gian topo mờ phân bậc. Khi đó đối với mỗi giá trị $r \in [0,1]$, họ các tập mờ $\tau_r = \{\underline{A} \in I^X : \tau_r(\underline{A}) \geq r\}$ là topo mờ kinh điển.

Lân cận của một điểm trong không gian (X, τ): Một tập mờ $\underline{A} \in I^X$ được gọi là một lân cận α của điểm $x \in X$ nếu $\exists U \in I^X$ sao cho $\tau(U) > 0$, $\alpha < \mu_U(x)$ và $U \subseteq \underline{A}$.

Khái niệm cơ sở lân cận và cơ sở của không gian topo mờ được định nghĩa tương tự như trong trường hợp topo kinh điển.

Định lý 4: Một họ $\mathcal{B} = \{B \in I^X : \tau_r(B) \geq \mathbf{0}\}$ là một cơ sở của không gian topo mờ (X, τ) khi và chỉ khi với mỗi $U \in I^X$ sao cho $\tau(U) > \mathbf{0}$, U có thể biểu diễn như là hợp (trong đại số các tập mờ) của một số các phần tử nào đó trong $\mathcal{B}$.

2.2 Giải tích mờ (Fuzzy Analysis)

Như trên đã trình bày, mọi lý thuyết toán học đều có thể được mở rộng sang các lý thuyết toán học mờ. Trong phần này chúng ta trình bày một vài lý thuyết như vậy.

2.2.1 Phương trình vi phân mờ

Để đơn giản chúng ta xét trường hợp phương trình vi phân bậc một. Trong giải tích kinh điển nó có dạng

$$\frac{dy}{dt} = f(t, y, k) \qquad \text{với điều kiện ban đầu } y(0) = c. \qquad \text{(D1)}$$

trong đó k là vectơ n hằng số, t là biến trên một đoạn đóng giới nội chứa giá trị 0, $c \in \mathscr{R}$, còn y và f là các vectơ.

Bây giờ ta tìm cách mở rộng (D1) thành phương trình vi phân mờ (xem M. Friedman, M. Making and A. Kandel: Numerical solutions of fuzzy differential and integral equations, Vol.106, N1(1999); J.J. Buckley and T. Feuring: Fuzzy differential equations; J.Y. Park and H.K. Han: Fuzzy differential equations, Vol.110, N.1 (2000)).

Trước hết ta đưa ra khái niệm số mờ.

Định nghĩa 3: (Số mờ) Một tập mờ $\tilde{A}$ được gọi là số mờ nếu nó là tập mờ trên trường số thực $\mathscr{R}$ và thoả mãn các điều kiện sau:

a) $\exists x_0 \in \mathscr{R}$ sao cho $\mu_{\tilde{A}}(x_0)=1$, trong đó $\mu_{\tilde{A}}(x)$ là hàm thuộc của tập mờ $\tilde{A}$.

b) Hàm $\mu_{\tilde{A}}$ liên tục từng khúc trên $\mathscr{R}$.

Việc cần thiết là định nghĩa các phép tính 'số học' trên các số mờ và một quan hệ bất đẳng thức giữa chúng. Nhìn chung việc định nghĩa các phép toán và các phép so sánh các số mờ (Ranking Fuzzy Sets) là một vấn đề phức tạp. Nhằm đơn giản các định nghĩa này chúng ta giới hạn chỉ xét các số mờ có dạng sau đây và gọi là $L-R$ số mờ. Chúng sẽ có dạng hình học giống hình thang với hai cạnh bên được thay bằng các đường cong đơn điệu.

Gọi L (Left) và R (Right) là hai hàm tham chiếu, tức là hàm thoả mãn các tính chất:

1) $L(x)=L(-x)$.
2) $L(0) = 1$.
3) L không tăng trên $[0,+\infty)$. Khi đó $L-R$ số mờ là tập mờ với hàm thuộc có dạng:

$$\mu_{\tilde{A}}(x) = \begin{cases} L((A^L - x)/\alpha & \text{nếu } x \le A^L, \alpha > 0 \\ R((x - A^U)/\beta & \text{nếu } x \ge A^U, \beta > 0 \\ 1 & \text{cho các trường hợp còn lại} \end{cases}$$

trong đó $A^L < A^U$ và $\{A^L, A^U\}$ được gọi là lõi (Core) của $\tilde{A}$, tức là $\mu_{\tilde{A}}(x)=1$, $\forall x \in [A^L, A^U]$ với A^L, A^U là các giá trị modal trên và dưới (theo nghĩa modal logic) của tập mờ $\tilde{A}$. Một số mờ như vậy được ký hiệu là $\tilde{A} = [A^L, A^U, \alpha, \beta]_{LR}$.

Một lớp quan trọng các $L-R$ số mờ là các số mờ hình thang (với các cạnh bên tuyến tính), ký hiệu là $(A^L, A^U, \alpha, \beta)$, vì tính đơn giản của các phép tính và dễ sử dụng trong ứng dụng thực tiễn đối với các nhà kỹ thuật.

Cho hai số mờ hình thang bất kỳ, $\tilde{a} =(a^L, a^U, \alpha, \beta)$, $\tilde{b} =(b^L, b^U, \gamma, \theta)$. Các phép tính trên các số mờ được định nghĩa như sau:

1) Nhân số mờ với một số thực:

   Với $x>0$, $x\in R$: $x\tilde{a} = (xa^L, xa^U, x\alpha, x\beta)$.

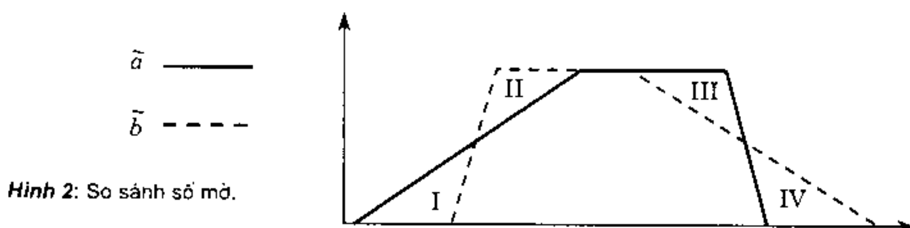
   Với ó $x<0$, $x\in R$: $x\tilde{a} = (xa^L, xa^U, -x\alpha, -x\beta)$.

2) Tổng và hiệu hai số mờ:

   $\tilde{a} + \tilde{b} =(a^L+b^L, a^U+b^U, \alpha+\gamma, \beta+\theta)$.

   $\tilde{a} - \tilde{b} =(a^L-b^L, a^U-b^U, \alpha+\gamma, \beta+\theta)$.

3) Ta quy ước rằng ứng với mỗi số mờ $\tilde{a}$ ta gán một số thực $r(\tilde{a}) = a^L+a^U+\dfrac{1}{2(\beta-\alpha)}$, gọi là phần tử đại diện của $\tilde{a}$.



Hình 2: So sánh số mờ.

4) So sánh hai số mờ: Giả sử hai số mờ đã cho có biểu diễn như trong hình trên. Ký hiệu S_i, i=I, II, III, IV tương ứng là diện tích của các miền I, II, III, IV và kí hiệu:

   $C(\tilde{a}, \tilde{b}) = S_{II} - S_I + S_{III} - S_{IV}$.

   Khi đó ta nói $\tilde{a} \geq \tilde{b}$ nếu và chỉ nếu $C(\tilde{a}, \tilde{b}) \geq 0$.

Bây giờ ta có thể chuyển phương trình vi phân thông thường (D1) thành phương trình vi phân mờ như sau:

Gọi $\tilde{K} = (\tilde{K}_1, \ldots, \tilde{K}_n)$ là một vectơ các số mờ hình thang (mỗi $\tilde{K}_i$ là một số mờ hình thang), và $\tilde{C}$ là một số mờ hình thang. Thay thế các giá trị này vào phương trình (D1) ta thu được một phương trình vi phân mờ:

$$\frac{d\tilde{Y}}{dt} = f(t, \tilde{Y}, \tilde{K}), \quad \tilde{Y}(0) = \tilde{C}. \qquad \text{(D2)}$$

2.2.1.1 Chúng ta khảo sát bài toán với điều kiện sau: Giả sử rằng $\widetilde{Y}(t) = [y_1(t), y_2(t)]$ và $y_i(t, \alpha)$, $i = 1, 2$, là hàm khả vi theo t với α là tham số. Ký hiệu đạo hàm của $y_i(t, \alpha)$ theo t là $y_i'(t, \alpha)$.

Đặt $G(t, \alpha) = [y_1'(t, \alpha), y_2'(t, \alpha)]$. Nếu $G(t, \alpha)$ chính là một lát cắt α của một số mờ thì ta nói hàm mờ $\widetilde{Y}(t)$ khả vi và được viết:

$$\frac{d\widetilde{Y}}{dt}[\alpha] = G(t, \alpha) = [y_1'(t, \alpha), y_2'(t, \alpha)]. \tag{D3}$$

Một điều kiện đủ để cho $G(t, \alpha)$ là lát cắt (α–cut hay α–level) của một số mờ là:

a) $y_1'(t, \alpha)$, $y_2'(t, \alpha)$ là các hàm liên tục theo cả hai biến.

b) $y_1'(t, \alpha)$ là hàm tăng theo biến α.

c) $y_2'(t, \alpha)$ là hàm giảm theo biến α.

d) $y_1'(t, \alpha) \le y_2'(t, \alpha)$, điều kiện này đảm bảo $[y_1'(t, \alpha), y_2'(t, \alpha)]$ là một đoạn thẳng.

Tất nhiên $\widetilde{Y}(t)$ là nghiệm nếu $\frac{d\widetilde{Y}}{dt}$ tồn tại và chúng thoả các đẳng thức trong (D2). Từ đó suy ra $\widetilde{Y}(t)$ là nghiệm nếu $\frac{d\widetilde{Y}}{dt}$ tồn tại và ta có các đẳng thức sau:

a) $y_1'(t, \alpha) = f_1(t, \alpha)$, $\quad y_2'(t, \alpha) = f_2(t, \alpha)$.

b) $y_1'(0, \alpha) = c_1(\alpha)$, $\quad y_2'(0, \alpha) = c_2(\alpha)$,

trong đó $\widetilde{C}(\alpha) = [c_1(\alpha), c_2(\alpha)]$.

2.2.1.2 Các phép đạo hàm của hàm mờ

Như trên ta đã nói, nhìn chung có nhiều cách mở rộng khái niệm vi phân mờ. Dưới đây ta đưa ra một số cách định nghĩa đạo hàm mờ (giá trị của hàm là tập mờ).

Giả sử $\widetilde{X}(t)$ nhận giá trị số mờ và giả sử $\widetilde{X}(t)[\alpha] = [x_1(t, \alpha), x_2(t, \alpha)]$ và $x_i'(t, \alpha)$ là đạo hàm riêng theo t của $x_i(t, \alpha)$.

Xét hai hàm mờ $\widetilde{X}(t)$ và $\widetilde{Z}(t)$. Ký hiệu lát cắt α của hai số mờ $\widetilde{X}(t)$ và $\widetilde{Z}(t)$ là

$$\widetilde{X}(t)[\alpha] = [x_1(t, \alpha), x_2(t, \alpha)],$$

và $$\widetilde{Z}(t)[\alpha] = [z_1(t, \alpha), z_2(t, \alpha)].$$

Gọi $D(\widetilde{X}(t), \widetilde{Z}(t))$ là metric giữa hai số mờ được định nghĩa một cách xác định nào đó. Khi đó đại lượng

$$\tilde{X}(t_0) = \lim_{h \to 0} \frac{\tilde{X}(t_0 + h) - \tilde{X}(t_0)}{h},$$

nếu tồn tại. là đạo hàm của hàm mờ $\tilde{X}(t)$.

1) Nếu $D(\tilde{X}(t), \tilde{Z}(t)) = \sup_{\alpha} \{ |x_1(t,\alpha) - z_1(t,\alpha)|, |x_2(t,\alpha) - z_2(t,\alpha)| \}$, ta có đạo hàm Goetschel-Voxman và ký hiệu là $GVD\ \tilde{X}(t_0)$. trong đó biểu thức hiệu trong công thức dưới lim được hiểu là hiệu số học theo từng thành phần toạ độ của vectơ $\tilde{X}$.

2) Nếu ta có hai điều kiện sau đây:

   a) $D(\tilde{X}(t), \tilde{Z}(t)) = \sup_{\alpha} H(\tilde{X}(t)[\alpha], \tilde{Z}(t)[\alpha])$. trong đó H là khoảng cách Hausdoff giữa các tập compact của R.

   b) Biểu thức hiệu trong công thức dưới lim là hiệu Hukuhara, tức là hiệu hai số mờ H A và B, được kí hiệu là $A-B$, là một số mờ C sao cho $B \oplus C = A$, trong đó $\oplus$ là phép cộng trên số mờ .

   thì khi đó ta có đạo hàm Puri-Relescu và kí hiệu là $PRD\ \tilde{X}(t_0)$.

3) Nếu ta có:

   a) $$D_p(\tilde{X}(t), \tilde{Z}(t)) = \max\left\{ \left(\int_0^1 |x_1(t,\alpha) - z_1(t,\alpha)|^p d\alpha \right)^{\frac{1}{p}}, \left(\int_0^1 |x_2(t,\alpha) - z_2(t,\alpha)|^p d\alpha \right)^{\frac{1}{p}} \right\}$$

   với tích phân lấy trên các hàm trong $L_p[0,1]$.

   b) Hiệu trong biểu thức dưới lim được hiểu như trong trường hợp 1).

   thì khi đó ta có đạo hàm Kandel-Friedman-Minh và kí hiệu là $KFMD\ \tilde{X}(t_0)$.

Định lý 5: (1) Nếu đạo hàm $GVD\ \tilde{X}(t)$ tồn tại và là một số mờ thì đạo hàm $SD\ \tilde{X}(t)$ cũng tồn tại và ta có $GVD\ \tilde{X}(t) = SD\ \tilde{X}(t)$.

(2) Nếu đạo hàm $PRD\ \tilde{X}(t)$ tồn tại và là một số mờ thì đạo hàm $SD\ \tilde{X}(t)$ cũng tồn tại và ta có $PRD\ \tilde{X}(t) = SD\ \tilde{X}(t)$.

(3) Nếu đạo hàm $KFMD\ \tilde{X}(t)$ tồn tại và là một số mờ thì đạo hàm $SD\ \tilde{X}(t)$ cũng tồn tại và chúng bằng nhau.

2.3 Bài toán tối ưu hoá mờ

Chúng ta đều biết đến sự phát triển về lý thuyết và ứng dụng của các bài toán tối ưu hoá trong đó có sự đóng góp đáng chân trọng của các nhà toán học Việt nam. Trên cách nhìn hệ thống hiện nay bài toán tối ưu luôn xuất phát từ nhu cầu của quá trình lấy quyết định và vì vậy nó gắn với việc lấy quyết định. Trong thực tế, quá trình này luôn luôn nằm trong môi trường các thông tin, dữ liệu không chính xác. biến động và bao gồm cả những dữ kiện có

bản chất không chắc chắn, mờ. Ngoài ra việc lấy quyết định thường nằm trong một hệ thống có cấu trúc thứ bậc và vì vậy nó cũng dẫn đến bài toán tối ưu nhiều mức (Multi-levels).

Như vậy thực tiễn đòi hỏi cần phát triển lý thuyết các bài toán tối ưu linh hoạt, mềm dẻo hơn. Cùng với sự phát triển lý thuyết toán học mờ, các bài toán tối ưu mờ cũng được nghiên cứu ứng dụng rất mạnh mẽ. Các bài toán tối ưu mờ có ý nghĩa thực tiễn lớn vì trong thực tế các ràng buộc không đòi hỏi quá chặt chẽ. Chẳng hạn việc giới hạn các ràng buộc bất đẳng thức theo nghĩa toán học kinh điển (ví dụ tổng các hạng mục đầu tư đòi hỏi không vượt quá 20.000 tỷ đồng) sẽ có thể loại đi mất những `nghiệm` cho kết quả tốt hơn (chẳng hạn với nó tổng vốn đầu tư có thể bằng 20.000 tỷ đồng cộng 1xu !!).

Có thể nói một đặc trưng quan trọng của phương pháp luận mờ là tính linh hoạt, mềm dẻo trong một môi trường phức tạp với thông tin, dữ kiện không chắc chắn, không chính xác và biến động, hơn nữa bài toán tối ưu mờ có thể được ứng dụng ngay đối với cả các bài toán kinh điển tuy giải được nhưng bằng những thuật toán quá phức tạp. Về nguyên tắc, sẽ xuất hiện các dạng bài toán tối ưu rất khác nhau tuỳ theo nhu cầu thực tế và quan điểm xây dựng mô hình.

Dưới đây chúng tôi chỉ giới thiệu đôi dạng bài toán (xem Vol.55,N3(1993)285-294; Vol.109, N1(2000)3-20,21-34,141-147).

2.3.1 Dạng bài toán tối ưu hoá với dữ kiện mờ

Giả sử các dữ kiện tham số trong bài toán tối ưu mờ được giới hạn trong lớp các số mờ.

Định nghĩa 4: (mô hình tối ưu hoá tuyến tính với số mờ) Gọi $F(R)$ là tập tất cả các số mờ hình thang. Mô hình bài toán tối ưu hoá tuyến tính với số mờ có dạng sau:

$$\max \tilde{z} = \sum_{j=1}^{p} \tilde{c}_j x_j ,$$

với các ràng buộc $\sum_{j=1}^{p} \tilde{a}_{ij} x_j \le \tilde{b}_i$, $i = 1,2, \dots, m_0$,

và $\sum_{j=1}^{p} \tilde{a}_{ij} x_j \ge \tilde{b}_i$, $i = m_0+1, \dots, m$, $j = 1,2, \dots, p$, trong đó $\tilde{a}_{ij}, \tilde{b}_i, \tilde{c}_j \in F(R)$.

Dạng bài toán này có thể chuyển về bài toán tối ưu thông thường nhờ mệnh đề sau.

Mệnh đề 1: Bài toán tối ưu hoá mờ tuyến tính trên tương đương với bài toán tối ưu sau:

$\max z = \sum_{j=1}^{p} r(\tilde{c}_j) x_j$, với các ràng buộc $\sum_{j=1}^{p} r(\tilde{a}_{ij}) x_j \le s(\tilde{b}_i)$, $i = 1,2, \dots, m_0$, và $\sum_{j=1}^{p} r(\tilde{a}_{ij}) x_j \ge s(\tilde{b}_i)$, $x_j \ge 0$, $i = m_0+1, \dots, m$, $j = 1,2, \dots, p$, trong đó r và s là hàm số mà giá trị của chúng được tính bằng một biểu thức trên các thông số của số mờ.

2.3.2 Bài toán tối ưu hoá tuyến tính với biến mờ

Bài toán này được phát biểu như sau.

Định nghĩa 5: Bài toán tìm nghiệm tối thiểu sau (sau đây gọi là bài toán A)

min: $\tilde{Z} = b'\tilde{Y}$

với các ràng buộc: $\tilde{Y}A \geq \tilde{C}$, $\tilde{Y} \geq 0$,

trong đó $0 \leq b \in R$, $A \in R^{m \times n}$. $\tilde{Y}$ gọi là bài toán tối ưu tuyến tính với biến mờ.

Định nghĩa 6: Bài toán hỗ trợ (gọi là bài toán B) là bài toán

max: $\tilde{Z} = \tilde{C}X$

với các ràng buộc: $AX < b, X \geq 0$,

trong đó $0 \leq b \in R^m$, $X \in R^m$, $A \in R^{m \times n}$, $\tilde{C} \in (F(R))^n$, $\tilde{Y} \in (F(R))^n$.

Giữa hai bài toán trên có mối quan hệ chặt chẽ thể hiện trong mệnh đề sau.

Định lý 6: (1) Nếu $\tilde{Y}^0$ là nghiệm *mờ* chấp nhận được của bài toán A và X^0 là nghiệm chấp nhận được của bài toán B, thì $\tilde{C}X^0 \leq b'\tilde{Y}^0$.

(2) Nếu $\tilde{Y}^0$ là nghiệm *mờ* chấp nhận được của bài toán A và X^0 là nghiệm chấp nhận được của bài toán B sao cho $\tilde{C}X^0 = b'\tilde{Y}^0$ thì X^0 là nghiệm tối ưu của bài toán B còn $\tilde{Y}^0$ là nghiệm tối ưu *mờ* của bài toán A.

(3) Nếu bài toán B có một nghiệm tối ưu thì bài toán A cũng có một nghiệm tối ưu *mờ*.

2.3.3 Bài toán quy hoạch nguyên mờ

Bài toán quy hoạch nguyên thông thường có rất nhiều ứng dụng thực tiễn như trong lập kế hoạch sản xuất chẳng hạn. Trong trường hợp như vậy các tham số là những đơn giá chi phí, nhu cầu vật tư hay bán thành phẩm và nguồn cung cấp (các bạn hàng cung cấp). Tuy nhiên trong thực tế thông tin về nhu cầu và khả năng cung cấp của các bạn hàng thường không chính xác, không chắc chắn. Tính không chính xác của những dữ liệu này còn là hệ quả của tính mềm dẻo, linh hoạt trong hoạch định sách lược, chiến lược của nhà máy. Thực tế đó đòi hỏi phải nghiên cứu ứng dụng bài toán quy hoạch nguyên mờ. Về trực cảm có thể thấy bài toán quy hoạch nguyên mờ sẽ rất gần gũi với cách giải bài toán kinh điển.

Chúng ta sẽ giới hạn tính mờ trong lớp các $L-R$ số mờ dạng $\tilde{A} = [A^L, A^U, \alpha, \beta]_{LR}$ đã được nói đến ở phần trên, nhưng ở đây L và R được thay thế bằng hàm số F thoả mãn điều kiện sau: F liên tục và không tăng trên nửa đường thẳng $[0,\infty)$, $F(0)=1$ và thực sự giảm trên miền mà F nhận giá trị dương.

Định nghĩa 7: Bài toán quy hoạch nguyên mờ được phát biểu như sau:

$c(x) = \sum_{i}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \rightarrow \min^*$, với các ràng buộc $\sum_{j=1}^{n} x_{ij} \cong A_i$, $i = 1,2, \ldots , m$ và $\sum_{i=1}^{m} x_{ij} \cong B_j$, $j = 1,2, \ldots , n$, $x_{ij} \geq 0$, $j = 1,2, \ldots , n$, $i = 1,2, \ldots , m$ và A_i và B_j là các số mờ. Các c_{ij} là chi phí vận chuyển được biểu thị bằng các giá trị số (không mờ). Đặc biệt min* được hiểu là mục tiêu mờ, hay một số mờ có dạng $[-\infty, c_0, 0, \beta_G]_{LR}$.

Trong môi trường mờ, chúng ta cần hiểu các ràng buộc mờ và mục tiêu mờ G sẽ có mức độ thoả nào đó (degree of satisfaction) và được định nghĩa như sau:

Định nghĩa 8: Giả sử x là một lời giải của bài toán. Khi đó

a) Giá trị được biểu thị bởi biểu thức sau

$$\mu_c(x) = \min\{ \mu_{A_i}\left(\sum_{j=1}^{n} x_{ij}\right)(i = 1, \cdots ,m), \mu_{B_j}\left(\sum_{i=1}^{m} x_{ij}\right)(j = 1, \cdots ,n), \}$$

được gọi là *độ thoả* của các ràng buộc;

b) Còn giá trị $\mu_G(x) = \mu_c(c(x)) = \mu_G\left(\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}\right)$,

được gọi là *độ thoả* của mục tiêu của bài toán quy hoạch nguyên mờ.

Bài toán A: Lời giải tối đại của bài toán quy hoạch nguyên mờ được định nghĩa là giá trị x sao cho hàm $\mu_D(x) = \min\{ \mu_c(x),\ \mu_G(x) \}$ *đạt cực đại*. Trong trường hợp giá trị cực đại này bằng 0 thì ta nói bài toán không khả thi.

Bài toán A tương đương với bài toán dưới đây.

Bài toán B: $\lambda \rightarrow \max$ với các ràng buộc sau

a) $\mu_G(c(x)) \geq \lambda$, $i = 1,2, \ldots , m$,

b) $\mu_{A_i}\left(\sum_{j=1}^{n} x_{ij}\right) \geq \lambda$, $i = 1,2, \ldots , m$,

c) $\mu_{B_j}\left(\sum_{i=1}^{m} x_{ij}\right) \geq \lambda$, $j = 1,2, \ldots , n$, $\lambda > 0$, $x_{ij} \geq 0$.

Gọi A_i^{λ} , B_j^{λ} và G^{λ} tương ứng là các lát cắt λ của các số mờ A_i, B_j và G. Khi đó bài toán B chuyển về bài toán sau.

Bài toán C: $\lambda \rightarrow \max$ với các ràng buộc sau

a) $c(x) \in G^{\lambda}$.

b) $\sum_{j=1}^{n} x_{ij} \in A_i^{\lambda}$, $i = 1,2, \ldots , m$,

c) $\sum_{i=1}^{m} x_{ij} \in B_j^{\lambda}, j = 1,2, \dots, n, \lambda>0, x_{ij}\geq 0.$

Như vậy bài toán quy hoạch nguyên mờ từng bước được đưa về bài toán quy hoạch nguyên thông thường (xem S. Chanas và D. Kuchta: Fuzzy integer transportation problem, Vol.98, N3(1998)) để tìm nghiệm bài toán gốc.

3 Hệ chuyên gia mờ và hệ trợ giúp quyết định mờ

3.1 Bài toán lấy quyết định và vấn đề lập luận

Một đặc trưng rất khác biệt của con người là khả năng lấy quyết định. Việc lấy quyết định là hoạt động diễn ra hàng ngày của mỗi người, của mỗi nhóm người và nó là hoạt động đặc biệt quan trọng trong lĩnh vực tổ chức và quản lý như việc ra nghị quyết, chính sách chế độ, lập kế hoạch, ra chỉ thị ....

Để dễ hiểu chúng ta sẽ chỉ ra những thành tố quan trọng nhất của quá trình lấy quyết định.

1) *Cơ sở (kho tàng) tri thức*: Như chúng ta đã biết con người khác biệt với loài vật ở trí tuệ, khả năng tư duy. Vì vậy một thành tố quan trọng đầu tiên của quá trình lấy quyết định là tri thức và được mô hình hóa thành một cơ sở tri thức. Các yếu tố cơ bản của tri thức có thể phát biểu thành các mệnh đề và dạng mệnh đề quan trọng nhất là dạng mệnh đề "nếu ... thì ..." và được gọi là các *luật* (rules).

Ví dụ: Trong lĩnh vực quản lý xí nghiệp cơ sở tri thức có thể chứa mệnh đề sau, trong đó các từ viết nghiêng là các khái niệm mờ: "Nếu sản phẩm của một xí nghiệp chiếm một thị phần *tương đối lớn* thì xí nghiệp đó *có thể* có chính sách điều phối giá cả".

Các chuyên gia trong lĩnh vực nghiên điều khiển mô tơ điện có thể phát biểu tri thức của mình bằng các mệnh đề if-then sau, trong đó I là cường độ dòng điện, N là tốc độ vòng quay của mô tơ (xem Cao, Z. and A. Kandel, Applicability of some fuzzy implication operators, Fuzzy Sets and Systems 31(1989),151-186), (xin sử dụng thuật ngữ tiếng Anh để có chỗ có thể dễ hiểu hơn):

If I = very small then N = very large
If I = very more small then N = large
If I = small then N = medium
If I = medium then N = small
If I = large then N = very more small
If I = very large then N = very more small.
v.v...

2) *Cơ sở dữ liệu* (CSDL): Có thể thấy tri thức là những khẳng định đã được tổng kết, khái quát hoá từ kinh nghiệm thực tiễn. Kinh nghiệm này được "bộ óc" lưu trữ ở dạng dữ

liệu (biểu thị ở dạng chữ hay số). Vì vậy một thành tố khác trong quá trình lấy quyết định là tập hợp các dữ liệu được tổ chức thành cơ sở dữ liệu. Trong các bài toán lớn như việc xây dựng các hệ trợ giúp quyết định, CSDL là một thành tố quan trọng vì hai lý do:

- Thứ nhất, nó lưu trữ các dữ liệu cần thiết cho quá trình lấy quyết định.
- Thứ hai, vì dữ liệu là kinh nghiệm thực tiễn nên kho dữ liệu này là cơ sở để điều chỉnh và phát hiện thêm các luật mới của tri thức.

Một hướng quan trọng đang được quan tâm nghiên cứu là việc phát hiện luật từ một tập hợp dữ liệu. Ví dụ từ một tập hợp dữ liệu của hơn hai trăm loài hoa người ta có thể xây dựng thuật toán cho phép tìm ra 7 hoặc 8 luật mờ về mối quan hệ giữa đặc tính màu sắc của hoa, số đài hoa, số cánh hoa và tính chất của lông quang đài và cuống hoa v.v....

3) *Phương pháp, thủ tục lập luận*: Thành tố quan trọng nhất của quá trình lấy quyết định là vấn đề lập luận. Bản chất phương pháp tư duy, lập luận của con người vẫn còn là một vấn đề bí hiểm, phức tạp và chúng ta chưa hiểu biết được nhiều. Nhiều phương pháp lập luận để mô phỏng phương pháp lập luận của con người đã và đang được nghiên cứu. Những phương pháp mô phỏng dựa trên logic toán đã được nghiên cứu phát triển từ rất sớm nhưng chúng chỉ phù hợp cho các ứng dụng dựa trên khoa học chính xác, dựa trên thông tin đầy đủ và chắc chắn, ví dụ như phương pháp chứng minh tự động trong lĩnh vực toán học. Tuy nhiên sức mạnh tư duy và lập luận của con người lại nằm trong lĩnh vực (môi trường) thông tin không đầy đủ, mờ và không chắc chắn, không chính xác. Ví dụ trong môi trường như vậy chúng ta vẫn phải quyết định tuyển chọn cán bộ, chọn phương án phát triển, xây dựng khu vực công nghệ cao v.v....

Trong môi trường như vậy "công nghệ tập mờ", và hiện nay nó được khái quát thành khái niệm "công nghệ tính toán mềm", đang giữ một vai trò độc tôn.

3.2 Phương pháp lập luận xấp xỉ dựa trên tập mờ

Như trên đã trình bày, phương pháp lập luận là một thành tố rất đặc trưng của quá trình lấy quyết định và vì vậy một chức năng quan trọng của một hệ chuyên gia hay một hệ trợ giúp quyết định là việc lập luận cho phép rút ra những kết luận trợ giúp người lấy quyết định. Những phương pháp lập luận trong những hệ như vậy có khả năng mô phỏng quá trình lập luận trong môi trường thông tin không đầy đủ, không chắc chắn và vì vậy bản chất của phương pháp là xấp xỉ gần đúng. L.A. Zadeh vẫn là người đầu tiên đưa ra ý tưởng và xây dựng phương pháp luận này và gọi là phương pháp lập luận xấp xỉ (xem L. A. Zadeh: A theory of approximate reasoning, in: R. R. Yager, S. Ovchinnikov, R. M. Tong and H. T. Nguyen, Eds., *Fuzzy Sets and Applications: The selected papers by L. A. Zadeh* (Wiley, New York, 1987) 367-411; L. A. Zadeh: The concept of linguistic variable and its application to approximate reasoning (I),(II), *Inform. Sci.* **8**(1975) 199-249; **8**(1975) 310 - 357).

Trong công trình của mình, Zadeh đưa ra khái niệm lược đồ lập luận xấp xỉ như sau:

Tiên đề 1:	Nếu *màu* của quả cà chua nào đó là *đỏ* thì quả cà chua đó là *chín*.
Tiên đề 2:	Màu quả cà chua Q là *rất đỏ*.
Kết luận:	Quả cà chua là *rất chín*.

Chúng ta thấy lược đồ này tương tự như luật Modus ponens trong logic kinh điển: từ $A \Rightarrow B$ và A cho phép ta rút ra kết luận B. Tuy nhiên ở lược đồ trên trong giả thiết (tiền đề) ta không có A mà lại có A' (:= rất đỏ) một biến chướng của A (:= đỏ), và mỗi người trong chúng ta đều có khả năng rút ra một kết luận B' nào đó. Vấn đề là cần xây dựng phương pháp luận cho phép lựa chọn phương pháp lập luận để "tính" B' sao cho kết quả phù hợp với ứng dụng cụ thể đã cho.

Nhờ tính mềm dẻo của phương pháp luận tập mờ chúng ta có nhiều phương án lựa chọn để xây dựng phương pháp lập luận xấp xỉ. Trong phạm vi bài này, chúng tôi chỉ trình bày phương pháp đơn giản nhất qua đó để bạn đọc thấy được ý tưởng của phương pháp luận.

Chúng ta xét lược đồ lập luận mờ đa điều kiện, t.l. mô hình mờ có chứa nhiều mệnh đề điều kiện dạng nếu…thì:

Tiên đề 1	if $X=A_1$ then $Y=B_1$
Tiên đề 2	if $X=A_2$ then $Y=B_2$
⋮	⋮
Tiên đề n	if $X=A_n$ then $Y= B_n$
Tiên đề $n+1$	if $X=A_{n+1}$ then $Y= B_{n+1}$
Kết luận:	$Y=B_0$

Tập hợp n mệnh đề đầu tiên trong (M) được gọi là *mô hình mờ*, trong đó A_i , B_i là các khái niệm mờ. Mô hình này mô tả mối quan hệ (hay sự phụ thuộc) giữa hai đại lượng X và Y. Giá trị $X=A_0$ được gọi là input còn $Y=B_0$ được gọi là output của mô hình.

Phương pháp lập luận xấp xỉ tính $Y=B_0$ gồm các bước sau:

1) **Bước 1.** *Giải nghĩa các mệnh đề điều kiện*: Chúng ta xem các khái niệm mờ A_i,B_i là nhãn của các tập mờ biểu thị ngữ nghĩa của A_i,B_i. Để cho tiện hàm thuộc của chúng được kí hiệu tương ứng là $A_i(u)$ và $B_i(u)$ trên các không gian tham chiếu U và V.

   Một cách trực cảm, mỗi mệnh đề nếu…thì trong mô hình mờ có thể hiểu là một phép kéo theo (implication operator) trong một hệ logic nào đó và được viết $A_i(u) \Rightarrow B_i(u)$. Khi u và v biến thiên, biểu thức này xác định một quan hệ mờ $R_i : U \times V \to [0, 1]$. Như vậy mỗi mệnh đề điều kiện trong (M) xác định một quan hệ mờ.

2) **Bước 2.** *Kết nhập* (aggregation) các quan hệ mờ thu được bằng công thức:

$R = @_{i=1}^{n} R_i$, trong đó @ là một phép tính *t–norm* hay *t–conorm* nào đó.

Chẳng hạn $R = \wedge_{i=1}^{n} R_i$ hay $R = \vee_{i=1}^{n} R_i$, trong đó $\wedge, \vee$ là các phép tính min và max.

Việc kết nhập như vậy đảm bảo *R* chứa thông tin được cho bởi các mệnh đề if…then có trong mô hình mờ.

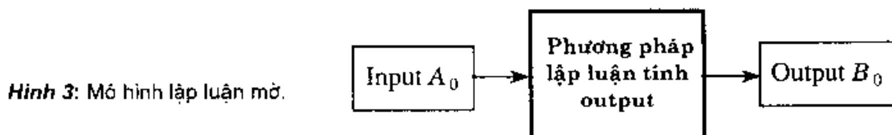
3) **Bước 3**. *Tính output* B_0 theo công thức $B_0 = A_0 o R$, trong đó o là phép hợp thành giữa hai quan hệ A_0 và R.

4) **Bước 4**. *Khử mờ* (Defuzzification: xem Vol.55,N1(1993)1-14; Vol.80,N2(1996)177-186; Vol.81,N3(1996)259-262): Kết quả tính toán ở bước 3 là một tập mờ. Trong nhiều bài toán ứng dụng, đặc biệt trong điều khiển kỹ thuật, người ta cần biết giá trị thực của biến *Y*. Phương pháp tính giá trị thực "tương ứng" với tập mờ B_0 được gọi là phương pháp khử mờ. Người ta đưa ra nhiều phương pháp khử mờ. Sẽ không có phương pháp nào gọi là tốt nhất. Nó sẽ được lựa chọn để đạt được "độ tốt" qua thử nghiệm. Ta lấy ví dụ một phương pháp khử mờ, gọi là phương pháp khử mờ theo trung bình cộng có trọng số, được cho bởi công thức:

$$\text{defuz}(B_0) = \frac{\sum_{v \in V} v B_0(v)}{\sum_{v \in V} B_0(v)}.$$

Những yếu tố ảnh hưởng đến kết quả tính toán của phương pháp lập luận mờ:

Có thể hình dung phương pháp lập luận mờ bằng mô hình tổng quát sau:



Hình 3: Mô hình lập luận mờ.

Chúng ta nhận thấy có nhiều phương pháp lập luận mờ. Mỗi phương pháp đều phụ thuộc vào:

– việc chọn các hàm thuộc dùng để biểu diễn ngữ nghĩa của các khái niệm mờ,
– việc chọn toán tử kéo theo để tính các quan hệ mờ R_i,
– việc chọn phương pháp kết nhập,
– việc chọn phép tính hợp thành o,
– và cuối cùng phụ thuộc vào chính phương pháp khử mờ.

Hiện nay chưa có phương pháp nào hỗ trợ việc lựa chọn này ngoài trực cảm, kinh nghiệm và qua thử nghiệm. Nhiều khi sử dụng các phép toán có ý nghĩa đối nghịch nhau nhưng nó vẫn góp phần hoàn thiện kết quả.

Thuật toán xấp xỉ ngôn ngữ:

Tuy nhiên, trái với sự đòi hỏi ở trên, trong các ứng dụng khác, chẳng hạn như trong việc trợ giúp người lấy quyết định, chúng ta lại đòi hỏi kết quả của bài toán phải được biểu diễn ở dạng ngôn ngữ. Như vậy kết quả của phương pháp lập luận mờ ở dạng tập mờ (t.l. *một hàm số*) cần phải được chuyển thành dạng ngôn ngữ. Đây là một bài toán đòi hỏi các thuật toán phức tạp và thời gian tính toán lớn.

Chúng ta có thể hình dung bài toán như sau: Để dễ hiểu chúng ta xét các giá trị chân lý ngôn ngữ gồm *true, false, very true, very false, more-or-less true, more–or–less false, possibly true, possibly false, approximately true, approximately false, little true, little false, very possibly true, very possibly false, ...* Có thể xem các dữ liệu này như là nhãn của các tập mờ, hay nói một cách khác, chúng ta có thể biểu diễn ngữ nghĩa của các dữ liệu này bằng các tập mờ (*các hàm số*) mà chúng ta cho là thích hợp đối với ứng dụng đang được xem xét. Như vậy chúng ta có một *họ các hàm số* $\mathcal{K}$ biểu diễn ngữ nghĩa của các dữ liệu ngôn ngữ.

Khi đó bài toán xấp xỉ ngôn ngữ có thể phát biểu nôm na như sau: *Cho một hàm bất kỳ B_0 (là tập mờ - chẳng hạn là output của bài toán lập luận mờ), hãy tìm một hàm F trong họ $\mathcal{K}$ sao cho nó nằm gần B_0 đã cho nhất.*

Như vậy chúng ta có thể cảm nhận thấy thực sự lý thuyết tập mờ cung cấp cho chúng ta một cách tiếp cận tính toán cho phép mô phỏng quá trình lập luận của con người.

3.3 Đại số gia tử và lập luận xấp xỉ

Đại số gia tử và ứng dụng của chúng trong lập luận xấp xỉ đã được Phòng Nghiên cứu các Hệ Trợ giúp Quyết định và Hệ Chuyên gia, Viện Công nghệ Thông tin, nghiên cứu phát triển từ hơn chục năm nay(xem tài liệu liệt kê trong phần *Lập luận trên cơ sở đại số gia tử*). Trong bài tổng quan này chúng tôi chỉ trình bày những ý tưởng chính của cách tiếp cận này.

Như trên chúng tôi đã trình bày, để xây dựng phương pháp luận tính toán nhằm giải quyết vấn đề mô phỏng tư duy, lập luận của con người chúng ta phải thiết lập ánh xạ (t.l. phép nhúng theo thuật ngữ toán học) gán mỗi khái niệm mờ một tập mờ trong không gian tất cả các hàm $F(U,[0,1])$. Nghĩa là chúng ta mượn cấu trúc tính toán rất phong phú của tập $F(U,[0,1])$ để mô phỏng phương pháp lập luận của con người thường vẫn được thực hiện trên nền ngôn ngữ tự nhiên.

Vậy một vấn đề đặt ra là liệu bản thân ngôn ngữ có cấu trúc tính toán không? Nếu có thì các phương pháp lập luận xây dựng trên đó đem lại những lợi ích gì?

Trả lời cho câu hỏi thứ nhất, chúng tôi chỉ ra rằng tập các giá trị ngôn ngữ của một biến ngôn ngữ, tức là biến mà giá trị của nó được lấy trong miền ngôn ngữ, sẽ là một cấu trúc đại số đủ giàu để tính toán.

Chúng ta sẽ làm sáng tỏ điều này qua việc xét miền ngôn ngữ (linguistic domain) của biến chân lý TRUTH gồm các từ sau: *dom(TRUTH)* = {*true, false, very true, very false, more-or-less true, more-or-less false, possibly true, possibly false, approximately true, approximately false, little true, little false, very possibly true, very possibly false, ...*}, trong đó *true, false* là các từ nguyên thủy, các từ nhấn (modifier hay intensifier) *very, more-or–less, possibly, approximately, little* gọi là các gia tử (hedges).

Khi đó miền ngôn ngữ *T=dom(TRUTH)* có thể biểu thị như là một đại số $AT=(T,G,H,\leq)$, trong đó G là tập các từ nguyên thuỷ được xem như là các phần tử sinh, H là tập các gia tử được xem như là các phép toán một ngôi, quan hệ ≤ trên các từ (các khái niệm mờ) là quan hệ thứ tự được "cảm sinh" từ ngữ nghĩa tự nhiên. Ví dụ dựa trên ngữ nghĩa, các quan hệ thứ tự sau là đúng: *false ≤ true, more true ≤ very true* nhưng *very false ≤ more false, possibly true ≤ true* nhưng *false ≤ possibly false*, v.v.... T được sinh ra từ G bởi các phép tính trong H. Như vậy mỗi phần tử của T sẽ có dạng biểu diễn là $x=h_n h_{n-1} h_1 u,\ u \in G$. Tập tất cả các phần tử được sinh ra từ một phần tử x được ký hiệu là $H(x)$. Nếu G chỉ gồm có đúng hai từ *nguyên thuỷ mờ*, thì một được gọi là phần tử sinh dương ký hiệu là t, một gọi là phần tử sinh âm ký hiệu là f và ta có $f < t$. Trong ví dụ trên *True* là dương còn *False* là âm.

Hai phần tử của đại số gia tử được gọi là đối nghịch nhau nếu chúng có dạng biểu diễn với cùng một dãy các gia tử nhưng phần tử sinh của chúng khác nhau, t.l. một cái là dương, một cái là âm. Một đại số gia tử là đối xứng nếu mỗi phần tử chỉ có ***duy nhất*** một phần tử đối nghịch.

Kết quả tổng quát chính của chúng tôi về đại số gia tử có thể được phát biểu trong mệnh đề sau:

Định lý 7: Có tồn tại một hệ tiên đề hoá sao cho mỗi miền ngôn ngữ AT của biến ngôn ngữ trở thành dàn đầy đủ (complete lattice) có một phần tử 0, một phần tử đơi vị 1 và một phần tử trung hoà. Như vậy phép tuyển $\cup$ và hội $\cap$ logic có thể định nghĩa được trong cấu trúc này. Hơn nữa, nếu AT là một đại số gia tử đối xứng thì trong cấu trúc đó ta có thể định nghĩa phép phủ định – và phép kéo theo $\Rightarrow$ và ta có:

a) $-hx=h-x$, với mọi $h \in H$.

b) $- -hx=x$, $-1=0$, $-0=1$ và $-W=W$.

c) $-(x \cup y)=(-x \cap -y)$ và $-(x \cap y)=(-x \cup -y)$.

d) $x \cap -x \leq y \cup -y$ với mọi $x, y \in X$.

e) $x \cap -x \leq W \leq y \cup -y$;

f) $x > y$ khi và chỉ khi $x < -y$.

g) $x \Rightarrow y = -x \Rightarrow -y$.

h) $x \Rightarrow (y \Rightarrow z) = y \Rightarrow (x \Rightarrow z)$.

i) $x \Rightarrow y \geq x' \Rightarrow y'$ khi và chỉ khi $x \leq x'$ và/hoặc $y \geq y'$.

j) $1 \Rightarrow x=x$, $x \Rightarrow 1=1$, $0 \Rightarrow x=1$ và $x \Rightarrow 0 \leq -x$.

k) $x \Rightarrow y \geq W$ khi và chỉ khi hoặc $x \leq W$ hoặc $y \geq W$.

l) $x \Rightarrow y \leq W$ khi và chỉ khi $y \leq W$ và $x \geq W$.

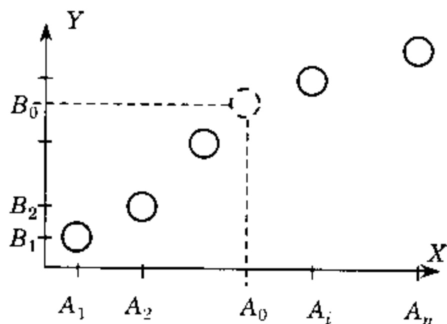
m) $x \Rightarrow y = 1$ khi và chỉ khi hoặc $x = 0$ hoặc $y = 1$.

Kết quả trên chứng tỏ rằng cấu trúc đại số gia tử đủ giàu để nghiên cứu các phương pháp lập luận.

1) *Lập luận xấp xỉ dựa trên đại số gia tử:*

Xét mô hình mờ (*M*). Vì chúng ta có thể xem các miền ngôn ngữ *X* và *Y* như là các đại số gia tử, một cách trực cảm ta có thể hình dung mỗi mệnh đề if…then trong (*M*) sẽ xác định một điểm và do đó mô hình (*M*) sẽ xác định cho ta một đường cong trong không gian ngôn ngữ $X \times Y$ và gọi là đường cong mờ *C*. Khi đó bài toán lập luận xấp xỉ trên tập mờ có thể chuyển về bài toán nội suy đối với đường cong mờ *C* (hình 4).

Để giải bài toán xấp xỉ, trước hết chúng ta phải lượng hoá đại số gia tử bằng cách trang bị cho nó một metric. Vì trong điều khiển học, các đại lượng *X* và *Y* thường là các đại lượng vật lý với giá trị lấy trên đường thẳng nên việc cho một metric trên *X* hay *Y* tương đương với việc cho một ánh xạ *f* từ *X* hay *Y* vào đoạn thẳng [0.1] với sự sai khác một hệ số tỷ lệ (vì trong thực tế ta cần ánh xạ vào một đoạn thẳng $[0, a]$ nào đó). Do đó ta đưa vào định nghĩa sau:



Hình 4: Lập luận xấp xỉ với đại số gia tử.

Định nghĩa 9: $f_s: X \rightarrow [0.1]$ gọi là hàm ngữ nghĩa định lượng của *X* nếu với mọi $h, k \in H^+$ hoặc $h, k \in H^-$ và mọi $x, y \in X$, ta có: $\left|\frac{f(hx) - f(x)}{f(kx) - f(x)}\right| = \left|\frac{f(hy) - f(y)}{f(ky) - f(y)}\right|$.

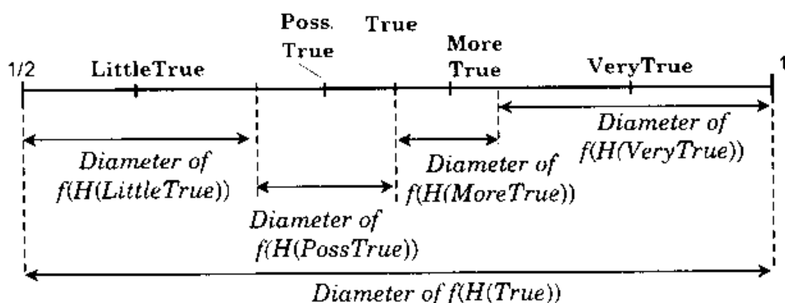
Với đại số gia tử và hàm ngữ nghĩa định lượng chúng ta có thể định nghĩa một khái niệm rất trìu tượng và khó định nghĩa một cách thoả đáng trong lý thuyết tập mờ là *tính mờ* của một khái niệm mờ hay của tập mờ biểu diễn nó.

2) *Tính mờ (Fuzziness) của một giá trị ngôn ngữ.*

Xét các giá trị: *True*, *Very False*, …. Làm thế nào định nghĩa tính mờ?

Trên quan điểm đại số gia tử ta có một cách định nghĩa *tính mờ* khá trực quan dựa trên kích cỡ của tập $H(x)$ như sau (hình 5).

Cho trước một hàm định lượng ngữ nghĩa *f* của *X*. Xét bất kỳ $x \in X$. *Tính mờ của x khi đó được đo bằng đường kính của tập* $f_s(H(x)) \subseteq [0,1]$.



Hình 5: Tính mờ của giá trị ngôn ngữ.

Định nghĩa 10: *Độ đo tính mờ*. Hàm $fm: X \to [0,1]$ được gọi là độ đo tính mờ nếu:

a) $fm(c^-)=w>0$ và $fm(c^+)=1-w>0$, c^-

trong đó c^- và c^+ là các phần tử sinh âm và dương.

b) Giả sử tập các gia tử là $H=H^+ \cup H^-$, $H^-=\{h_1, h_2, \dots h_p\}$ với $h_1>h_2> \dots >h_p$. và $H^+=\{h_{p+1}, h_{p+2}, \dots h_{p+q}\}$ với $h_{p+1}<h_{p+2}< \dots <h_{p+q}$. Khi đó, $\sum_{i=1}^{p+q} fm(h_i c) = fm(c)$ với $c \in \{c^+, c^-\}$.

c) Với bất kỳ $x,y \in X$, $h \in H$, $\frac{fm(hx)}{fm(x)} = \frac{fm(hy)}{fm(y)}$, t.l. đẳng thức này không phụ thuộc vào các phần tử x,y và do đó ta có thể kí hiệu là $\mu(h)$ và gọi là độ đo tính mờ (*fuzziness measure*) của gia tử h.

Mệnh đề 2 (*Tính chất của $fm(x)$ và $\mu(h)$*). Chúng ta có:

a) $fm(hx)= \mu(h)fm(x), \forall\ x \in X$.

b) $\sum_{i=1}^{p+q} fm(h_i c) = fm(c)$, $c \in \{c^+, c^-\}$.

c) $\sum_{i=1}^{p+q} fm(h_i x) = fm(x)$.

d) $\sum_{i=1}^{p} \mu(h_i) = \alpha$ và $\sum_{i=p+1}^{p+q} \mu(h_i) = \beta$ với $\alpha, \beta > 0$ và $\alpha + \beta = 1$.

3) *Xây dựng hàm định lượng ngữ nghĩa trên cơ sở độ đo tính mờ của gia tử.*

Định nghĩa 11: Hàm sgn : $X \to \{-1, 0, 1\}$

a) $\operatorname{sgn}(c^-)=-1$ và $\operatorname{sgn}(hc^-)=\begin{cases} +\operatorname{sgn}(c^-) & \text{nếu} \quad hc^- < c^- \\ -\operatorname{sgn}(c^-) & \text{nếu} \quad hc^- \ge c^- \end{cases}$.

b) $\text{sgn}(c^+)=+1$ và $\text{sgn}(hc^+)=\begin{cases}+\text{sgn}(c^+) & \text{nếu} \quad hc^+ < c^+ \\ -\text{sgn}(c^+) & \text{nếu} \quad hc^+ \geq c^+\end{cases}$.

c) $\text{sgn}(h'hx)=-\text{sgn}(hx)$ nếu h' là negative đ/với h và $h'hx \neq hx$.

d) $\text{sgn}(h'hx)=-\text{sgn}(hx)$ nếu h' là positive đ/với h và $h'hx \neq hx$.

e) $\text{sgn}(h'hx)=0$ nếu $h'hx=hx$.

4) *Xây dựng hàm định lượng ngữ nghĩa*: Giả sử cho trước độ đo tính mờ của các gia tử $\mu(h)$, và các giá trị độ đo tính mờ của các phần tử sinh $fm(c^-)$, $fm(c^+)$ và w là phần tử trung hoà (Neutral).

Hàm định lượng ngữ nghĩa v của X được xây dựng như sau với $x=h_{im}\ldots h_{i2}h_{i1}c$:

a) $v(c^-)=w-\alpha fm(c^-)$, $v(c^+)=w+\alpha fm(c^+)$.

b) $fm(x)=fm(h_{im}\ldots h_{i2}h_{i1}c)=\mu(h_{im})\ldots\mu(h_{i2})\mu(h_{i1})fm(c)$.

c) $$v(h_jx)=v(x)+\text{sgn}(h_jx)\left[\sum_{i=j}^{p}fm(h_ix)-\frac{1}{2}(1-\text{sgn}(h_ix)\text{sgn}(h_1h_ix)(\beta-\alpha)fm(h_ix)\right]$$

nếu $j<p$ và

$$v(h_jx)=v(x)+\text{sgn}(h_jx)\left[\sum_{i=p+1}^{j}fm(h_ix)-\frac{1}{2}(1-\text{sgn}(h_ix)\text{sgn}(h_1h_ix)(\beta-\alpha)fm(h_ix)\right]$$

nếu $j>p$.

5) *Giải bài toán lập luận mờ bằng nội suy.*

Xét mô hình mờ (*M*). Giả sử $\mathcal{X}$, $\mathcal{Y}$ là các đại số gia tử sinh ra từ các giá trị ngôn ngữ tương ứng xuất hiện trong mô hình. Kí hiệu $\mathcal{C}$ đường cong mờ trong không gian $\mathcal{X}\times\mathcal{Y}$.

Giả sử $fs_{\mathcal{X}}$ và $fs_{\mathcal{Y}}$ là các hàm định lượng ngữ nghĩa tương ứng của $\mathcal{X}$ và của $\mathcal{Y}$. Các hàm này sẽ chuyển đường cong mờ $\mathcal{C}$ thành đường cong thực C trong không gian $[0,1]\times[0,1]$.

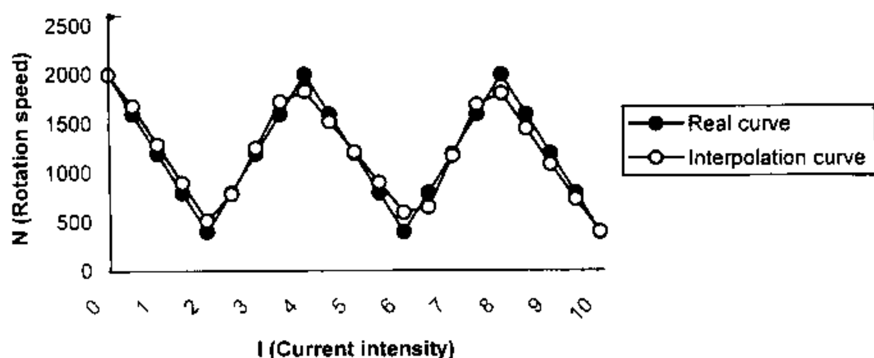
Như vậy bài toán lập luận mờ được chuyển về bài toán nội suy thông thường nhờ hàm định lượng đại số gia tử.

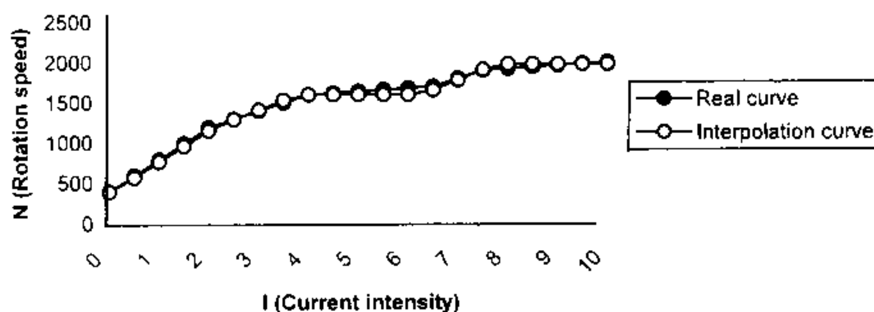
Có thể thấy phương pháp này có một số ưu điểm sau:

- Cho một ý tưởng trực quan rõ ràng về cách thức giải bài toán.
- Trong phương pháp giải dựa trên lý thuyết tập mờ có rất nhiều yếu tố gây sai số như: xây dựng hàm thuộc; chọn cách giải nghĩa mệnh đề if-then bằng quan hệ mờ (thực chất là chọn việc giải nghĩa toán tử kéo theo); chọn toán tử kết nhập (Aggregation) các quan hệ; chọn phép hợp thành để tính output; chọn phương pháp khử mờ và khó có được trực giác trong việc xây dựng phương pháp giải. Còn trong phương pháp nội suy dựa trên đại số gia tử chúng ta chỉ phải tập trung nỗ lực vào việc lựa chọn độ đo tính mờ của các

gia tử và chúng trở thành hệ tham số của phương pháp. Vì vậy nó rất gần gũi với các cách giải kinh điển.

- Không cần phương pháp khử mờ! Lưu ý rằng trong lý thuyết tập mờ có khá nhiều phương pháp khử mờ.
- Thực nghiệm kiểm chứng đã chỉ ra rằng nó cho sai số nhỏ (hình 6).





Hình 6: Giải bài toán lập luận mờ bằng nội suy.

3.4 Hệ trợ giúp quyết định mờ

Ngay sau khi máy tính điện tử ra đời nó đã được nhanh chóng ứng dụng vào việc xây dựng các hệ xử lý thông tin và sau này, cùng với sự phát triển, chúng được gọi là các hệ thông tin quản lý (MIS: Management Information System). Theo nghĩa rộng các hệ MIS cũng hàm chứa chức năng trợ giúp quyết định, chúng cung cấp các thông tin cần thiết cho nhận định tình hình hay phát hiện vấn đề. Tuy nhiên các hệ như vậy không có khả năng can

thiệp vào các quá trình lấy quyết định và đưa ra những phương án tốt cho người lấy quyết định.

3.4.1 Hệ trợ giúp quyết định mờ

Mục tiêu xây dựng các hệ trợ giúp quyết định và các hệ chuyên gia là làm sao chúng có khả năng sản sinh ra các phương án lấy quyết định hợp lý trước các tình huống cụ thể được phản ánh, cung cấp qua các dữ liệu. Các dữ liệu phản ánh tình huống và các thông tin được sản sinh ra trong các phương pháp và các quá trình lấy quyết định của người lấy quyết định thường hàm chứa các yếu tố không chính xác, không chắc chắn và biểu thị qua các dữ liệu ngôn ngữ. Vì vậy các hệ mờ sẽ có miền ứng dụng rộng lớn và có vai trò ứng dụng quan trọng, đi đôi với sự phát triển mạnh mẽ của xã hội hiện đại–một xã hội thông tin–tri thức.

Như chúng ta có thể cảm nhận, các quá trình lấy quyết định rất phức tạp, đa dạng và vì vậy cũng có rất nhiều phương pháp, mô hình xây dựng các hệ trợ giúp quyết định, trong đó yếu tố quan trọng nhất là các phương pháp lấy quyết định. Dưới đây chúng tôi trình bày một mô hình khá thông dụng để qua đó bạn đọc hình dung *phương pháp trợ giúp quyết định* và nhận diện *một phương pháp xử lý thông tin mờ*.

Giả sử người lấy quyết định sử dụng phương pháp chuyên gia, t.l. lấy ý kiến các chuyên gia, để lựa chọn phương án quyết định (chẳng hạn trong việc nên lựa chọn phương án xây dựng nhà máy lọc dầu như thế nào cho tối ưu, trong việc sinh viên tốt nghiệp nên lựa chọn làm việc ở đâu cho phù hợp, trong việc học sinh nên lựa chọn ngành học thế nào (công tác tư vấn hướng nghiệp), hay lựa chọn địa điểm, vùng đầu tư, ...). Các quá trình lấy quyết định như vậy có thể được mô hình hoá như sau.

Giả sử chúng ta có n khả năng lựa chọn (alternatives) $A_1, A_2, \ldots, A_n$. Hãy chọn khả năng tốt nhất trong n khả năng đã cho dựa trên ý kiến của m chuyên gia $E_1, E_2, \ldots, E_m$ đánh giá theo k tiêu chuẩn $C_1, C_2, \ldots, C_k$. Để phù hợp với thực tế, chúng ta giả thiết mỗi chuyên gia và mỗi tiêu chuẩn có một trọng số nào đó thể hiện mức độ tin cậy vào chuyên gia và tầm quan trọng của tiêu chuẩn, và ý kiến đánh giá của các chuyên gia có thể được biểu thị bằng dữ liệu ngôn ngữ (khái niệm mờ như *cao*, *thấp*, *trung bình*...) hay bằng số.

Chúng ta giới hạn các khái niệm mờ được biểu thị bằng số mờ dạng $\tilde{a} = (a^L, a^U, \alpha, \beta)$ và kí hiệu ý kiến của chuyên gia E_j đánh giá mức độ thoả mãn của khả năng lựa chọn A_i đối với tiêu chuẩn C_l là $\tilde{a}_{ij}^l = (a_{ij}^{Ll}, a_{ij}^{Ul}, \alpha_{ij}^l, \beta_{ij}^l)$.

Chúng ta cũng giả sử rằng tầm quan trọng của các tiêu chuẩn cũng được xác định dựa trên việc lấy ý kiến chuyên gia và kí hiệu $\tilde{b}_{ij}^l = (b_{ij}^{Ll}, b_{ij}^{Ul}, \delta_{ij}^l, \varepsilon_{ij}^l)$ $\tilde{b}_j^l = (b_j^{Ll}, b_j^{Ul}, \delta_j^l, \varepsilon_j^l)$ là ý kiến của chuyên gia E_j về tầm quan trọng của tiêu chuẩn C_l. Dựa vào các phép tính trên số mờ chúng ta thiết lập ý liến "trung bình cộng" của các chuyên gia bằng các biểu thức sau:

– Trung bình cộng các ý kiến của các chuyên gia về tầm quan trọng tiêu chuẩn C_l là:

$$\widetilde{d}^{\,l} = \frac{1}{n} \otimes \left(\widetilde{b}_1^l \oplus \widetilde{b}_2^l \oplus \cdots \oplus \widetilde{b}_n^l \right).$$

– Trung bình cộng các ý kiến của các chuyên gia về mức độ thoả mãn của khả năng lựa chọn A_i đối với tiêu chuẩn C_l là:

$$\widetilde{m}_i^l = \frac{1}{n} \otimes \left(\widetilde{a}_{i1}^l \oplus \widetilde{a}_{i2}^l \oplus \cdots \oplus \widetilde{a}_{in}^l \right).$$

– Cuối cùng, kết quả kết nhập các ý kiến của các chuyên gia về độ thoả mãn của khả năng lựa chọn A_i theo tất cả các tiêu chuẩn được cho bởi công thức:

$$\widetilde{\omega}_i = \frac{1}{kM} \otimes \left(\widetilde{m}_i^1 \oplus \widetilde{d}^{\,1} \oplus \widetilde{m}_i^2 \oplus \widetilde{d}^{\,2} \oplus \cdots \oplus \widetilde{m}_i^k \oplus \widetilde{d}^{\,k} \right).$$

trong đó M là một hằng số xác định.

– *Vấn đề sắp xếp các tập mờ*: Để chọn phương án tốt chúng ta phải so sánh (t.l. cố gắng sắp xếp chúng theo thứ tự tuyến tính) các kết quả $\widetilde{\omega}_i$, $i = 1,2, \ldots, n$. Chúng là những tập mờ. Như vậy chúng ta cần xây dựng các thuật toán sắp xếp các tập mờ (Ranking fuzzy sets). Đây cũng là một vấn đề không dễ có thuật toán tốt và hiện nay đối với bất kỳ thuật toán sắp xếp các tập mờ nào bao giờ cũng có những trường hợp không thể quyết định xem tập này có "lớn hơn" tập kia không và những thuật toán sắp xếp mới vẫn tiếp tục xuất hiện trên các tạp chí.

3.4.2 Các toán tử kết nhập

Ở trên chúng ta đã thấy, để kết quả đánh giá chung về các khả năng lựa chọn đều có phần ý kiến của các chuyên gia một cách hợp lý người ta đã sử dụng phép kết nhập (tích hợp) trung bình cộng có trọng số. Với nguyên tắc mềm dẻo của *công nghệ tính toán mềm* trong môi trường mờ, chúng ta lại thấy trong sự phát triển của lý thuyết tập mờ, người ta cần đưa ra nhiều phương pháp kết nhập (Aggregation) ý kiến chuyên gia khác nhau. Nói khác đi một phương pháp kết nhập không thể phù hợp cho mọi tình huống đa dạng, phức tạp khác nhau.

Một cách tiên đề một họ các toán tử kết nhập được giả thiết phải thoả một số điều kiện nào đó. Chẳng hạn tiên đề "bình đẳng" đòi hỏi khi các ý kiến của các chuyên gia là trùng nhau thì kết quả kết nhập phải trùng với ý kiến chung đó. Một lần nữa chúng ta lại có thể thấy sự đòi hỏi về tính mềm dẻo: không thể có một hệ tiên đề chung duy nhất cho các toán tử kết nhập. Tiên đề bình đẳng có thể phù hợp với những tình huống này nhưng không phù hợp với những tình huống khác, chẳng hạn không phù hợp đối với mô hình trợ giúp quyết định có gán trọng số cho các chuyên gia (không bình đẳng).

Về trực cảm chúng ta có thể thấy từ thực tiễn cuộc sống là việc kết nhập, tổng hợp ý kiến của mọi người là một việc phức tạp. Trên quan điểm toán học, những nhận xét trên thể

hiện không gian các dữ liệu biểu thị các ý kiến chuyên gia chỉ có *cấu trúc yếu* và do đó không thể đưa ra một định nghĩa toán tử kết nhập duy nhất.

Dưới đây chúng tôi xin trình bày một họ toán tử kết nhập gọi là uni-norm có cấu trúc toán khá chặt chẽ (xem Yager et al. FSS Vol.80(1996) []).

Định nghĩa 12: Toán tử uni–norm R là một ánh xạ hai ngôi $R:[0,1]\times[0,1]\rightarrow[0,1]$ thoả các tính chất sau:

a) $R(a,b) = R(b,a)$ – Tính chất giao hoán.

b) $R(a,b) \geq R(c,d)$ nếu $a \geq b$ và $b \geq d$ – Tính chất đơn điệu.

c) $R(a,R(b,c)) = R(R(a,b),c)$ – Tính chất kết hợp.

d) Tồn tại một phần tử đơn vị e nghĩa là với mọi $a \in [0,1]$ ta đều có $R(a,e)=a$.

Như vậy toán tử uni-norm tương tự như phép toán t–norm và t–conorm, trừ tính chất cuối cùng về phần tử đơn vị. Trong định nghĩa trên, phần tử đơn vị có thể khác các giá trị 0 và 1.

Định lý 8: Giả sử R là toán tử uni-norm với đơn vị là e. Khi đó

a) $\overline{R} = 1-R(\overline{a}, \overline{b})$, $\overline{a} = 1-a$ cũng là toán tử uni-norm với đơn vị là $1-e$.

b) $R(a,b) \geq a$, với mọi $b>e$ và $a \in [0,1]$.

c) $R(a,b) \leq a$, với mọi $b<e$ và $a \in [0,1]$.

d) $R(a_1, \ldots, a_n) \geq R(a_1, \ldots, a_n, a_{n+1})$, nếu $a_{n+1} < e$.

e) $R(a_1, \ldots, a_n) \leq R(a_1, \ldots, a_n, a_{n+1})$, nếu $a_{n+1} > e$.

f) $R(a,0) = 0$ với $\forall a \leq e$; $R(a,1) = 1$, với $\forall a \geq e$.

Mệnh đề sau đây chỉ ra sự tồn tại toán tử uni–norm:

Định lý 9:

I) Toán tử R_* được xác định như sau:

a) Với mọi a,b mà $\max(a,b) \leq e$ ta định nghĩa $R_*(a,b) = \min(a,b)$;

b) Với mọi a,b mà $\min(a,b) \geq e$ ta định nghĩa $R_*(a,b) = \max(a,b)$;

c) Với mọi a,b mà $\max(a,b) > e$ và $\min(a,b) < e$ ta định nghĩa $R_*(a,b) = \min(a,b)$,

là toán tử uni-norm với đơn vị là e.

II) Toán tử R_* được xác định như sau:

a) Với mọi a,b mà $\max(a,b) \leq e$ ta định nghĩa $R_*(a,b) = \min(a,b)$;

b) Với mọi a,b mà $\min(a,b) \geq e$ ta định nghĩa $R_*(a,b) = \max(a,b)$;

c) Với mọi a,b mà $\max(a,b) > e$ và $\min(a,b) < e$ ta định nghĩa $R_*(a,b) = \max(a,b)$,

là toán tử uni–norm với đơn vị là e.

Chúng ta có mệnh đề sau để đặc trưng thêm toán tử kết nhập uni-norm

Định lý 10:

1) Nếu $\min(a_1, \ldots, a_n) < e$ thì $R_*(a_1, \ldots, a_n) = \min(a_1, \ldots, a_n)$, trái lại thì $R_*(a_1, \ldots, a_n)$ $= \max(a_1, \ldots, a_n)$;

2) Nếu $\max(a_1, \ldots, a_n) > e$ thì $R_*(a_1, \ldots, a_n) = \max(a_1, \ldots, a_n)$, trái lại thì $R_*(a_1, \ldots, a_n)$ $= \min(a_1, \ldots, a_n)$.

4 Điều khiển mờ (Fuzzy Control)

4.1 Quá trình điều khiển với yếu tố mờ, không chắc chắn

Lý thuyết điều khiển đã được phát triển rất mạnh mẽ và đã tìm được những lĩnh vực ứng dụng rất rộng rãi trong công nghiệp, trong các hệ thông hoá học, các hệ không gian, ... Các phương pháp điều khiển truyền thống thường đòi hỏi người ta phải hiểu biết rõ bản chất của đối tượng cần điều khiển thông qua các mô hình toán học, và trong nhiều ứng dụng chúng thường là các phương trình toán lý phức tạp với bậc phi tuyến cao. Ngoài ra các đối tượng điều khiển thường nằm trong môi trường có những tác động gây nhiễu và người ta rất khó xác định được các đặc trưng. Những đối tượng phức tạp như vậy thường nằm ngoài khả năng giải quyết của các phương pháp điều khiển truyền thống và trong quá trình tự động hoá người ta phải nhờ vào khả năng xử lý tình huống của con người và phải thiết kế thiết bị sử dụng việc điều khiển bằng tay. Việc con người có khả năng điều khiển các quá trình như vậy chứng tỏ rằng các quá trình đó đã được phản ánh và mô phỏng đúng đắn bằng mô hình nào đó trong đầu óc của người điều hành. Như đã trình bày trong phần mở đầu, các mối quan hệ trong quá trình điều khiển này không phải được biểu thị bằng các mô hình toán lý mà bằng mô hình ngôn ngữ với các thông tin không chính xác, không chắc chắn hay nói khác đi, những thông tin mờ, có tính ước lệ hay định tính.

Mục tiêu cội nguồn của phương pháp điều khiển mờ chính là nhằm vào việc xây dựng các phương pháp có khả năng bắt chước cách thức con người điều khiển các quá trình với mô hình định tính như vậy trên cơ sở phương pháp luận lý thuyết tập mờ và công nghệ tính toán mềm. Việc bắt chước này có thể hiểu nôm na như sau:

Vì đối tượng điều khiển là một hệ phức tạp, có những bản chất chưa rõ và không thể biểu thị bằng các mô hình toán lý nên người "chuyên gia" điều hành hệ thống chỉ có thể quan sát thông tin vào – ra để phán đoán hành vi của hệ thống và trên cơ sở kinh nghiệm đó điều khiển hệ thống. Nhận thức về hành vi của hệ này được thâu tóm dưới dạng mô hình mờ (M), t.l. một tập các mệnh đề if…then (các luật) với các dữ liệu ngôn ngữ mô tả mối quan hệ giữa các biến vào, các biến ra. Việc mô phỏng hành vi của hệ trong thực tế điều khiển của người điều hành bằng một thuật toán tính được bằng máy tính chính là phương pháp lập luận mờ đa điều kiện đã trình bày ở trên.

Như vậy các phương pháp điều khiển mờ đều gắn với các phương pháp lập luận mờ và chúng có những đặc điểm sau:

1) Nó chỉ dựa trên các thông tin vào–ra quan sát được trên các đối tượng điều khiển, không đòi hỏi phải hiểu bản chất để mô hình hoá toán học đối tượng như trong lý thuyết điều truyền thống.
2) Mô hình định tính dựa trên ngôn ngữ (hay mô hình được biểu thị nhờ tri thức có tính chuyên gia): Thay vì phải mô hình hoá bằng các phương trình toán lý, phương pháp điều khiển mờ đòi hỏi phải thu thập được tri thức để thiết lập mô hình hoá định tính về đối tượng điều khiển. Tri thức này có thể thu thập từ các chuyên gia hay từ các thuật toán phân tích,khai thác dữ liệu mờ: phát hiện ra các luật, t.l. các mệnh đề if…then, từ đống dữ liệu quan sát được.
3) Giảm độ phức tạp tính toán nhờ mô hình định tính, tuy không có được tính chính xác mà mô hình toán học định lượng có được.
4) Miền ứng dụng rộng lớn đa dạng.

4.2 Phương pháp điều khiển mờ

Việc nghiên cứu ứng dụng điều khiển mờ hiện nay được phát triển rất mạnh mẽ. Ở nhiều nước công nghiệp phát triển, như Đức, Nhật …, một số phòng thí nghiệm chuyên nghiên cứu ứng dụng phương pháp luận mờ trong các lĩnh vực công nghiệp đã được thiết lập (như phòng thí nghiệm LIFE (Laboratory of International Fuzzy Engineering ở Nhật). Tính hiệu quả trong các ứng dụng kỹ thuật và công nghiệp đã hình thành tư tưởng, quan niệm thể hiện qua các thuật ngữ mới như công nghệ lý thuyết tập mờ (fuzzy sets technology), công nghệ logic mờ (fuzzy logic technology) và gần đây là công nghệ tính toán mềm (soft computing technology). Trong trào lưu như vậy tất nhiên sẽ ra đời rất nhiều phương pháp khác nhau để giải quyết các bài toán thức tế hàm chứa sự thông minh, trí tuệ mà phần lớn các bài toán như vậy có cấu trúc yếu hoặc phương pháp giải truyền thống quá phức tạp. Trong bài tổng quan về các hướng nghiên cứu ứng dụng của lý thuyết tập mờ này chúng tôi chỉ nêu tóm lược hai phương pháp điều khiển mờ khá "kinh điển".

4.2.1 Phương pháp xây dựng bộ điều khiển mờ dựa trên luật (Rule–based fuzzy controlers)

Bộ điều khiển mờ dựa trên luật lần đầu tiên được Mamdani và Assilan phát triển vào năm 1975 nhằm điều khiển tự động áp suất hơi nước và tốc độ cho Turbin phát điện thông qua việc thay đổi chế độ cung cấp nhiệt và van tiết lưu. Ý tưởng phương pháp thiết kế về tổng quát tương tự như đã trình bày ở trên và quan trọng là dựa vào việc xây dựng một phương pháp lập luận mờ và lượng hoá các khái niệm mờ cho phù hợp.

Đặc trưng của phương pháp này *là tập hợp các luật (tri thức) được xây dựng dựa vào tri thức của các chuyên gia, hay nói khác đi là tập luật là sự mô phỏng các tình huống đáp ứng của các chuyên gia trong quá trình thao tác điều khiển.*

Bộ điều khiển mờ kiểu như vậy đã được phát triển và ứng dụng trong nhiều công việc khác nhau, trong công nghiệp, trong điều khiển các lò xi măng v.v….

Phương pháp xây dựng kiểu bộ điều khiển này có ưu điểm là đơn giản và việc ứng dụng cũng khá phổ biến. Tuy nhiên nó cũng có những bất lợi. Thứ nhất, việc xây dựng tập hợp luật là một khó khăn. Các chuyên gia không dễ gì phát biểu tri thức của họ dưới dạng luật mặc dù họ có kinh nghiệm. Thứ hai, Một khi bộ điều khiển đã được xây dựng thì thuật toán điều khiển dựa trên tập luật đó là cố định, không thay đổi, không thể thích nghi theo sự thay đổi của quá trình điều khiển. Thứ ba là theo phương pháp thiết kế, bộ điều khiển sẽ cư xử giống như người chuyên gia điều khiển quá trình đó, mặc dù cư sử như vậy chưa chắc đã là tối ưu.

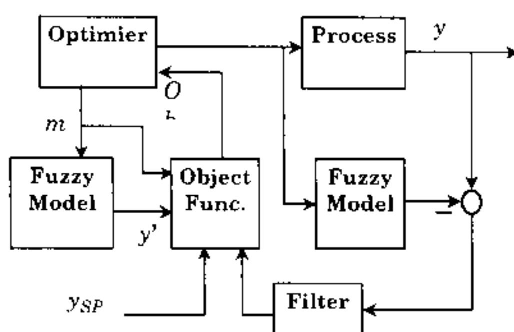
Những thiếu sót này dẫn đến việc phát triển các bộ điều khiển mờ tự tổ chức (self-organising–controller). Đặc điểm của bộ điều khiển kiểu này là nó sẽ tự sinh ra luật điều khiển của mình.

4.2.2 Phương pháp xây dựng bộ điều khiển mờ dựa trên mô hình (Model-based fuzzy controller)

Như trên đã trình bày ý tưởng chính của phương pháp trên là thuật toán điều khiển mô phỏng hành vi của người điều hành. Trái lại, ý tưởng của phương pháp này, tương tự như các phương pháp truyền thống, là mô hình hoá chính đối tượng hay quá trình điều khiển bằng mô hình mờ trên cơ sở khảo sát mối quan hệ giữa dữ liệu biến vào và biến ra. Mô hình hệ điều khiển có thể mô phỏng trong hình 7 dưới đây. Cách tiếp cận như vậy có những ưu điểm sau: 1) Về nguyên tắc việc khảo sát hành vi (sự đáp ứng) của đối tượng điều khiển thông qua inputs và outputs là dễ dàng hơn so với việc khảo sát sự đáp ứng của các chuyên gia (operators); 2) Phương pháp này cung cấp khả năng thiết kế mềm dẻo, linh hoạt hơn và cung cấp cách thức bộ điều khiển đạt được mục tiêu chứ không chỉ đơn giản là viếc đáp ứng lại các input; 3) Vẫn tận dụng lợi thế của phương pháp lập luận mờ.

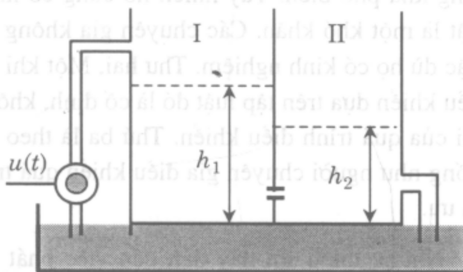
Những nội dung chính của phương pháp là:

- Xây dựng mô hình mờ quan hệ (relational fuzzy model) dựa trên quan sát input/output;
- Xây dựng thuật toán cho phép bộ điều khiển lựa chọn thao tác điều hành cho kết quả tốt nhất dựa trên hàm tổn thất mờ (fuzzy cost function);
- Có thể sử dụng mô hình tự học (self–learning) hay các mô hình lai giữa mô hình mờ và mô hình toán học.



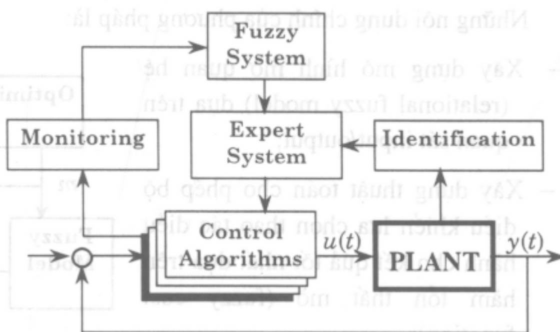
Hình 7: Điều khiển có mô hình mờ theo dõi.

4.2.3 Phương pháp điều khiển thông minh dựa trên tri thức và logic mờ



Hình 8: Điều khiển mực nước.

Trong thực tiễn điều khiển các quá trình người ta thường gặp những môi trường trong đó người ta không thể thu được các dữ liệu chính xác hoặc không thể mô tả chính xác trạng thái, …. trong những trường hợp như vậy cần phát triển, xây dựng các bộ điều khiển thông minh nhiều thành phần dựa trên việc kết hợp các kỹ thuật điều khiển chuyên gia để chọn các giải thuật điều khiển và sử dụng logic mờ để đánh giá hiệu năng thực hiện của giải thuật. Nghĩa là người ta sẽ sử dụng nhiều giải thuật điều khiển khác nhau trong cùng một bộ điều khiển. Một ví dụ đơn giản là việc điều khiển một hệ hai bể nước thông nhau. Nó là một hệ SISO (single-input – single–output). Tuy nhiên đây là một hệ phức tạp vì thực nghiệm chỉ ra rằng máy bơm có đặc trưng động học phi tuyến bậc cao liên quan đến điện áp đầu vào và tốc độ dòng chảy. Khi điện áp thấp máy bơm không tạo đủ áp suất đưa nước vào bể I, nhưng với điện áp cao, tốc độ dòng không phụ thuộc vào điện áp. Với tính phi tuyến như vậy người ta không thể áp dụng các bộ điều khiển thông thường để điều khiển quá trình này mà đòi hỏi phải ứng dụng phương pháp điều khiển thích nghi dựa trên cơ sở tri thức. Trong trường hợp này người ta sử dụng bốn giải thuật điều khiển tổng hợp khác nhau: bộ điều khiển tự điều chỉnh PID, bộ điều khiển PPC (Pole–placement controler), bộ điều khiển GMV (Generalised minimum variance controler), bộ điều khiển GPC (Generalised predictive controler).



Hình 9: Bộ điều khiển mờ thông minh.

Cấu trúc của bộ điều khiển thông minh được trình bày trong hình 9. Trong hình thể hiện hệ có chứa bốn giải thuật điều khiển mà sẽ được sử dụng theo một chiến lược dựa trên kiến thức (cơ sở tri thức) của chuyên gia.

Mặc dù phương pháp dựa trên công nghệ các hệ chuyên gia nhưng nó có những khác biệt đáng kể so với các hệ chuyên gia khác, như nó đòi hỏi phải sản sinh tín hiệu điều khiển trong thời gian thực (real time), không cần đòi hỏi sự tương tác với con người để thực hiện chức năng điều khiển (hệ chuyên gia hay hệ hỗ trợ quyết định, một cách tự nhiên, cần tương tác với người ra quyết định), bộ điều khiển thông minh cần phải tương tác trực tiếp với quá trình hay đối tượng điều khiển và cần trang bị các phương tiện để gắn bộ điều khiển vào quá trình ấy.

Bộ điều khiển loại này có những đặc điểm sau:

– Nó phải có khả năng sử dụng nhiều giải thuật điều khiển khác nhau, đánh giá được chiến lược điều khiển nào có khả năng sinh ra tín hiệu điều khiển thích ứng nhất và có khả năng điều chỉnh các thông số của mỗi giải thuật để thích ứng với những đòi hỏi trong các tình huống khác nhau. Việc đánh giá này dựa vào kỹ thuật tập mờ và logic mờ với dữ liệu ngôn ngữ như *nhỏ*, *trung bình*, *lớn* ....

– Nó có khả năng đánh giá và lựa chọn giải thuật để duy trì trạng thái gần tối ưu và đảm bảo độ tin cậy, và ngay cả khi một vài yếu tố của thiết bị điều khiển bị hỏng nó cũng có thể cấu hình lại giải thuật đó hay chuyển sang bộ giải thuật khác thích ứng hơn.

– Nó có một cơ sở tri thức về kinh nghiệm đối với quá trình điều khiển cùng với những tri thức dạng luật mà các giải thuật sẽ sử dụng trong quá trình điều khiển v.v....

Ví dụ đối với hệ hai bể nước các luật có thể là:

If Overshoot is *Small* AND Setting time is *Fast*, Then Score is *Large Positive*;

If Overshoot is *Small* AND Setting time is *Slow*, Then Score is *Medium Positive*;

If Overshoot is *Large* AND Setting time is *Optimal*, Then Score is *Small Positive*

v.v....

5 Tính toán mờ và tri thức

Tri thức có vai trò rất quan trọng trong các hoạt động "thông minh" của con người. Như chúng ta dễ dàng nhận thấy, mỗi con người dù là người nước nào đều tư duy bằng ngôn ngữ và như vậy tri thức trong đầu của mỗi người được biểu thị qua ngôn ngữ. Vì ngôn ngữ hàm chứa các thông tin mờ cho nên việc ứng dụng phương pháp luận mờ trong việc biểu diễn, xử lý, xây dựng các hệ tri thức là tự nhiên và như vậy mới thực sự đi vào giải quyết vấn đề một cách bản chất. Chính vì vậy các mô hình fuzzy được sử dụng để mô phỏng, phản ánh, giải quyết các bài toán liên quan đến các khía cạnh định tính của tri thức. Một trong những ứng dụng của công nghệ tính toán mờ là bài toán phát hiện, khai phá tri thức.

5.1 Khai phá dữ liệu (DM - Data Mining)

Cách đây vài năm chúng ta thấy rộ lên vấn đề khai thác dữ liệu (Data Mining (DM)) mà nội dung chủ yếu là phát hiện thông tin có tính trí tuệ trong kho tàng dữ liệu. Khai phá dữ liệu (DM) thường đi đôi với phát hiện tri thức (Knowledge Discovery (KD)), vì tuy mức độ "trí tuệ" của kết quả khai thác dữ liệu có khác nhau nhưng ý tưởng của chúng có chung một bản chất. Vì vậy trong mục này tuy chỗ này chỗ kia có sử dụng một trong hai thuật ngữ đó nhưng chúng ta hiểu đang nói cùng một vấn đề phương pháp luận.

Như chúng ta đã thấy, ngay sau khi ra đời vài năm, máy tính điện tử đã được ứng dụng vào lĩnh vực quản lý, một lĩnh vực hoạt động cần có thông tin và trí thức. Trong sự phát triển

của xã hội mà yếu tố khoa học công nghệ có tính quyết định nhu cầu như vậy trở nên ngày càng to lớn và dẫn đến sự bùng nổ thông tin và những nhà quản lý đứng trước tình trạng phải đối mặt với tình trạng lụt "thông tin". Ví dụ Hệ thông quan sát trái đất của NASA trong năm 1999 phải sinh tạo ra 50 Gb dữ liệu ảnh quan sát được mỗi giờ; thị trường Walmart phải thực hiện khoảng 20 triệu giao dịch dữ liệu mỗi ngày, ...

Chính vì vậy các chuyên gia cho rằng hiện nay chúng ta đang sống trong một xã hội rất giàu có về thông tin nhưng nghèo về thi thức. Tình hình đó đòi hỏi phải nghiên cứu phát triển các phương pháp khai phá, phát hiện ra những thông tin, tri thức hữu ích bị che dấu trong "đống" thông tin (dữ liệu) để phục vụ cho các công việc của các nhà quản lý, các chuyên viên, chuyên gia. Chẳng bao lâu nữa chúng ta sẽ chứng kiến sự gia tăng mạnh mẽ các ứng dụng của bài toán phát hiện tri thức từ dữ liệu.

Mục tiêu của bài toán khai phá dữ liệu là tìm ra thông tin, tri thức (t.l. những mối quan hệ hay sự phụ thuộc giữa các đại lượng (các biến)) trong kho tàng dữ liệu(về một lĩnh vực nào đó). Suy cho cùng *tri thức phản ánh mối quan hệ giữa các đại lượng mà quan hệ là cấu trúc của sự vật, hiện tượng.*

Bài toán cơ sở của vấn đề khai phá dữ liệu mờ có thể hình thức hoá như sau:

Cho một bảng dữ liệu với cột chỉ các thuộc tính (attribute) và mỗi hàng là một bộ dữ kiện. Bảng dũ liệu này sẽ phản ảnh những cái gì đó, hay nói cách khác nó hàm chứa những mối quan hệ bản chất nào đấy (tri thức) mà ta cần phải tìm, phát hiện ra. Chẳng hạn *T* là bảng dữ liệu về thời tiết với các thuộc tính về áp suất khí quyển, nhiệt độ, độ ẩm, tốc độ gió trung bình, cường độ bức xạ mặt trời, ...

Mỗi tri thức sẽ được phát hiện trên một góc nhìn nào đó, ta gọi là khung cảnh (context). Ở đây sẽ là khung cảnh mờ hay khung cảnh ngôn ngữ. Khung cảnh sẽ được kí hiệu là *C*. Ví dụ phát hiện tri thức trong bảng dữ liệu với khung cảnh được xác định bởi áp suất = small & nhiệt độ = medium.

Khi đó bài toán khai phá tri thức được phát biểu như sau:

Hãy tìm cấu trúc S trong bảng T với khung cảnh C trong đó C sẽ được biểu thị bằng tập mờ, ví dụ áp suất = small sẽ được biểu thị bằng một tập mờ P nhất định trong không gian tham chiếu chỉ số đo áp suất khí quyển.

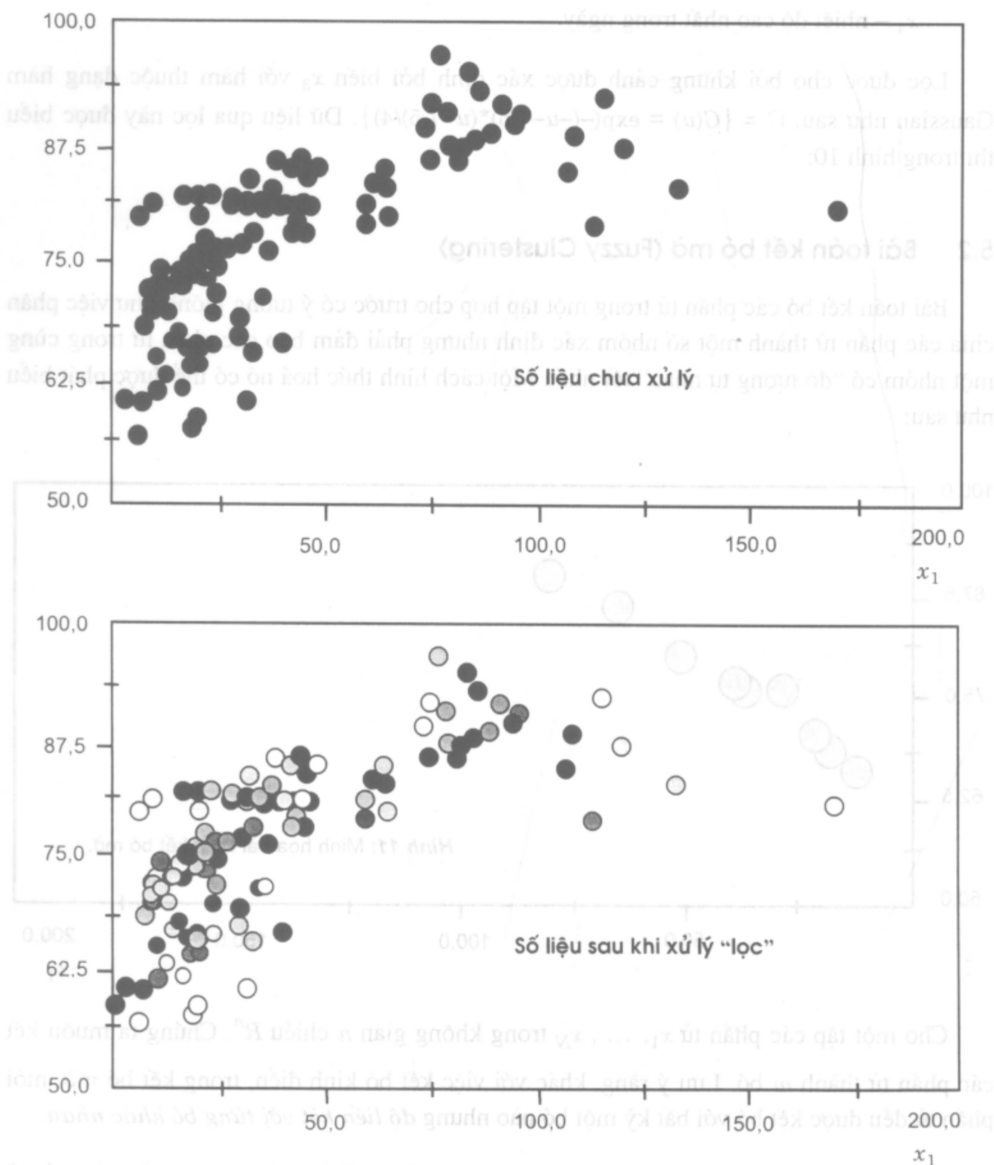
Trường hợp khung cảnh được xác định bởi nhiều biến, bài toán có thể phát biểu như sau:

Hãy tìm cấu trúc S của T với khung cảng C được xác được định bởi

{*áp suất = small* AND *nhiệt độ = medium*}.

Ta thấy ngay là mô hình mờ (khái niệm mờ hay ngôn ngữ) *rất dễ sử dụng, rất tự nhiên, cô đọng và trực cảm.* Lý thuyết tập mờ thiết lập cho chúng ta những (chứ không phải là một) cơ chế tính toán để giải bài toán này.

Có thể hình dung bài toán cơ bản này như là một cái "lọc" (filtering): dữ liệu qua cái lọc này sẽ cho kết quả là các dữ liệu thoả mãn những điều nào đó. Điều cần nhấn mạnh, đây là những điều kiện *mờ*.



Hình 10: Lọc mờ dữ liệu.

Ví dụ hãy xem xét đống dữ liệu đề cập đến sự trong lành của không khí vùng New York. Dữ liệu được thu thập trên cơ sở 4 thuộc tính:

x_1 – mức độ ozone,

x_2 – bức xạ mặt trời,

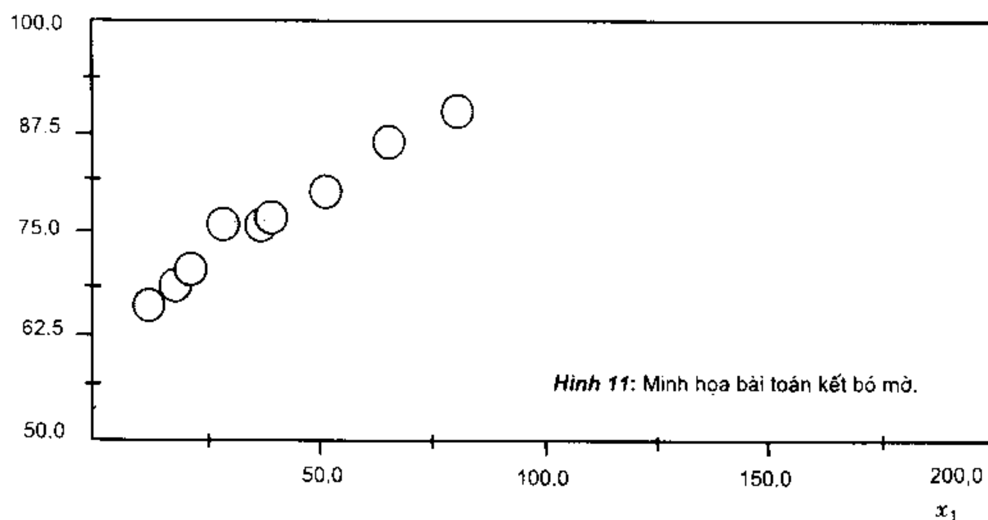
x_3 – tốc độ trung bình của gió,

x_4 – nhiệt độ cao nhất trong ngày.

Lọc được cho bởi khung cảnh được xác định bởi biến x_3 với hàm thuộc dạng hàm Gaussian như sau: $C = \{C(u) = \exp(-(-u-7,5)*(u-7,5)/4)\}$. Dữ liệu qua lọc này được biểu thị trong hình 10:

5.2 Bài toán kết bó mờ (Fuzzy Clustering)

Bài toán kết bó các phần tử trong một tập hợp cho trước có ý tưởng giống như việc phân chia các phần tử thành một số nhóm xác định nhưng phải đảm bảo các phần tử trong cùng một nhóm có "độ tương tự nhau" tốt nhất. Một cách hình thức hoá nó có thể được phát biểu như sau:



Hình 11: Minh họa bài toán kết bó mờ.

Cho một tập các phần tử $x_1, \ldots, x_N$ trong không gian n chiều R^n. Chúng ta muốn kết các phần tử thành m bó. Lưu ý rằng, khác với việc kết bó kinh điển, trong kết bó mờ, mỗi phần tử đều được kết bó với bất kỳ một bó nào nhưng *độ liên kết với từng bó khác nhau*.

Gọi z_i là phần tử mẫu của bó thứ i, $i = 1, \ldots, m$. Gọi $\boldsymbol{P}$ là họ các ma trận phân hoạch cỡ $m \times N$ được định nghĩa như sau:

$$\boldsymbol{P} = \{\mu = (pik): pik \in [0,1],\ \sum_{i=1}^{m} p_{ik} = 1,\ \forall k \text{ và } 0 < \sum_{i=1}^{m} p_{ik} < N \text{ với } \forall\, i\}.$$

Hãy kết bó sao cho nó làm tối tiểu hàm Q và ta có bài toán tối ưu sau:

$$\min_{\mu, x_1, \ldots, x_n} Q = \sum_{i=1}^{m} \sum_{k=1}^{N} p_{ik} |x_k - z_i|^2$$

với điều kiện ràng buộc là $\mu \in P$.

Để nhận dạng rõ hơn mối quan hệ của dữ liệu trong ví dụ trên, ta chỉ tập trung vào các phần tử mẫu được xác định qua sử dụng thuật toán kết bó cùng các kỹ thuật mờ khác và ta thu được đồ thị như ở hình 11:

6 Danh mục các tài liệu dẫn

Để bạn đọc thấy được sự phát triển của các hướng nghiên cứu và để tiện theo dõi lĩnh vực mình quan tâm chúng tôi liệt kê một số lượng khá lớn các tài liệu và sắp xếp theo các vực khoa học.

6.1 Giải tích

[1] **M. Friedman, M. Ming, A. Kandel:** On the validity of the Peano theorem for fuzzy differential equations, Fuzzy Sets and Systems, Vol. 86 N. 3,(1997)

[2] **J.Y.Park, H.K. Han, J.U. Jeong:** Asymptotic behaviour of solutions of fuzzy differential equations, Fuzzy Sets and Systems, Vol. 91 N.3 (1997).

[3] **F.M. Ali:** A differential equation approach to fuzzy non-linear programming problem, Fuzzy Sets and Systems Vol.93 N.1 (1998).

[4] **J.J. Buckley and T. Feuring:** Fuzzy differential equations, Fuzzy Sets and Systems, Vol.110 N.1 (2000).

6.2 Topo

[1] **A.S. Mashhour, M. H. Ghanim, A.N. EL- Wakeil and N.N.Morsi:** The productivity classes of fuzzy topologies, Fuzzy Sets and Systems Vol.53 N.2 (1993).

[2] **K.C Chattopadhyay and S.K. Samanta:** Fuzzy topology: Fuzzy closure operator, fuzzy compactness and fuzzy compactness. Fuzzy Sets and Systems Vol.54 N.2(1993).

[3] **Mingsheng Ying:** Compactness in fuzzifying topology, Fuzzy Sets and Systems, Vol5.5 N.1(1993).

[4] **Mingsheng Ying:** A new approach for fuzzy topology (III). Fuzzy Sets and Systems Vol.55 N.2(1993).

[5] **R. Srivastava:** On separation axioms in a newly defined fuzzy topology, Fuzzy Sets and Systems, Vol.62 N.3(1994).

[6] **P.Das:** Fuzzy topology on fuzzy sets: Product fuzzy topology and fuzzy topological groups, Fuzzy Sets and Systems, Vol.100 N.1-3(1998).

[7] **D. Adnadjevic:** Mappings and covering properties of topological groupoids, Fuzzy Sets and Systems, Vol.101 N.3(1999).

[8] **W. Gahler, A.S. Abd- Allah and A. Kandil:** On extended fuzzy topologies, Fuzzy Sets and Systems, Vol.109 N.2(2000).

[9] **V. Gregori and A. Vidal:** Gradations of openness and Chang fuzzy topologies, Fuzzy Sets and Systems, Vol.109 N.2(2000).

6.3 Bài toán quy hoạch

[1] **Herrera, J.L. Verdegay and H- J. Zimmermann:** Boolean programming problems with fuzzy constraints, Fuzzy Sets and Systems, Vol.55 N.3(1993).

[2] **H. Kuwano:** On the fuzzy multi- objective linear programming problems: goal programming approach, Fuzzy Sets and Systems, Vol.82 N.1(1996).

[3] **S. Chanas, D. Kuchta:** Fuzzy integer transportation problem, Fuzzy Sets and Systems, Vol.98, N.3(1998).

[4] **D. Wang:** An inexact approach for linear programming problems with fuzzy objective and resources, Fuzzy Sets and Systems, Vol.89, N.1(1997).

[5] **Y. Nakahara:** User oriented ranking criteria and its application topological fuzzy mathematical programming problems, Fuzzy Sets and Systems, Vol.94, N.3(1998).

[6] **M.A.E. Kassem:** A study of fuzzy parametric on multiobjective nonlinear programming problems without differentiability, Fuzzy Sets and Systems, Vol.98, N.1(1998).

[7] **S.M. Guu, Y.K. Wu:** Two-phase approach for solving the fuzzy linear programming problems, Fuzzy Sets and Systems, Vol.107, N.2(1999).

6.4 . Lý luận mờ và trợ giúp quyết định

[1] **Fuzzy Sets and Applications:** The selected papers by L. A. Zadeh (Wiley, New York, 1987), Eds: R. R. Yager, S. Ovchinnikov, R. M. Tong and H. T. Nguyen

[2] **L. A. Zadeh:** A theory of approximate reasoning, Inform. Sci. 8(1975)367-411.

[3] **L. A. Zadeh:** The concept of linguistic variable and its application to approximate reasoning (I), Inform. Sci. 8(1975)199-249.

[4] **L.A. Zadeh:** The concept of linguistic variable and its application to approximate reasoning (II), Inform. Sci. 8(1975) 310 -357.

[5] **E. Eslami, J.J. Buckley:** Inverse approximate reasoning, Fuzzy Sets and Systems Vol.87, N.2(1997).

[6] **T. Yamashita:** On a support system for human decision making by the combination of fuzzy reasoning and fuzzy structural modelling, Fuzzy Sets and Systems Vol.87, N.3(1997).

[7] **J.L. Castro, E. Trillas, J.M. Zurita:** Non-monotonic fuzzy reasoning, Fuzzy Sets and Systems Vol.94, N.2(1998).

[8] **Y. Shi, M. Mizumoto:** A note on reasoning conditions of Koczy's interpolative reasoning method (Short Communication), Fuzzy Sets and Systems Vol.96,N.3(1998).

6.5 Defuzzification

[1] **S. Mabuchi:** A proposal for a defuzzification strategy by the concept of sensitivity analysis, Fuzzy Sets and Systems Vol.55,N.1(1993).

[2] **R.R. Yager and D. Filev:** On the issue of defuzzification and selection based on a fuzzy set, Fuzzy Sets and Systems Vol.55,N.3(1993).

[3] **R.R. Yager:** Knowledge- based defuzzification, Fuzzy Sets and Systems Vol.80,N.2(1996).

[4] **L.Rondeau, R. Ruelas, L. Levrat, M. Lamotte:** A defuzzification method respecting the fuzzification, Fuzzy Sets and Systems Vol.86,N.3(1997).

6.6 Lập luận dựa trên đại số gia tử

[1] **N. Cat Ho:** Fuzziness in structure of linguistic truth values: A foundation for development of fuzzy reasoning, Proc. of ISMVL '87, Boston, USA (IEEE Computer Society Press, New York),1987, 326 - 335.

[2] **N. Cat Ho:** A method in linguistic reasoning on a knowledge base representing by sentences with linguistic belief degree, Fundamenta Informaticae Vol, 28 (3,4) (1996), 247-259 (also appeared at Linguistic-valued logic and a deductive method in linguistic reasoning, Proc. of the Fifth IFSA' 93, Seoul, Korea, July 4-9, 1993).

[3] **N. Cat Ho, H.V. Nam, T.D. Khang, N.H. Chau:** Hedge algebras, linguistic-valued logic and their application to fuzzy reasoning, Inter. J. of Uncertainty, Fuzziness and Knowledge-Based Syst., Vol.7, No.4(1999) 347-361.

[4] **N. Cat Ho & H. Van Nam:** A refinement structure of hedge algebras, Proc. of the NCST of Vietnam 9 (1) (1997), 15-28.

[5] **N. Cat Ho & H. Van Nam:** A theory of refinement structure of hedge algebras and its application to fuzzy logic, in: D. Niwinski & M. Zawadowski, Eds., Logic, Algebra and Computer Science, Banach Centre Publications (PWN-Polish Scientific Publishers), 1999, 63-91.

[6] **N. Cat Ho & W. Wechler**: Hedge algebras: An algebraic approach to structure of sets of linguistic truth values. Fuzzy Sets and Systems 35 (1990), 281-293.

[7] **N. Cat Ho & W. Wechler**: Extended hedge algebras and their application to fuzzy logic, Fuzzy Sets and Systems 52 (1992), 259 - 281.

6.7 Decision Making

[1] **L.-M. Jia and XX.-D. Zhang**: Distributed intelligent railway traffic control Based on fuzzy decision making, Fuzzy Sets and Systems Vol.62,N.3(1994).

[2] **F. Herrera, E. Herrera- Viedma and J.L. Verdegay**: A model of consensus in group decision making under linguistic assessments, Fuzzy Sets and Systems Vol.78,N.1(1996).

[3] **Ch. Carlsson and R. Fuller**: Fuzzy multiple criteria for decision making: Recent development, Fuzzy Sets and Systems, Vol.78,N.2(1996).

[4] **F. Herrera, E. Herrera- Viedma and J.L. Verdegay**: Direct approach processes in group decision making using linguistic OWA operator Fuzzy Sets and Systems, Vol.79,N.2(1996).

[5] **S. Bodjanova**: Approximation of fuzzy concepts in decision making, Fuzzy Sets and Systems, Vol.85,N.1(1997).

[6] **K. Meier**: Methods for decision making with cardinal numbers and additive aggregation, Fuzzy Sets and Systems. Vol.88,N.2(1997).

[7] **F. Herrera, E. Herrera Viedma, J.L. Verdegay**: Choice processes for non-homogeneous group decision making in linguistic setting, Fuzzy Sets and Systems, Vol. 94,N.3(1998).

[8] **K.K.Dompere**: Cost-benefit analysis, benefit accounting and fuzzy decisions (II). A hypothetical application topological mental illness, Fuzzy Sets and Systems. Vol.100, N.1-3(1998).

6.8 Điều khiển mờ

[1] **V. Novak**: Fuzzy control from the logical point of view, Fuzzy Sets and Systems, Vol.66,N.2(1994).

[2] **W. Z. Qiao and M. Mizumoto**: PID type fuzzy controller and parameters adaptive method, Fuzzy Sets and Systems, Vol.78,N.1(1996).

[3] **B.E. Postlethwaite**: Building a model- based fuzzy controller, Fuzzy Sets and Systems, Vol./79,N.1(1996).

[4] **Misir, H. A. Malki and G. Chen**: Design and analysis of a fuzzy proportional-integral- derivative controller, Fuzzy Sets and Systems, Vol.79,N.3(1996).

[5] **Tong Shaocheng, chai Tianyou, Zhang Huaguang**: Notes on multivariable fuzzy controller under Goedel implication, Fuzzy Sets and Systems, Vol.88, N.2(1997).

[6] **C.C. Fuh, P.C. Tung**: Robust stability analysis of fuzzy control systems, Fuzzy Sets and Systems, Vol.88. N.2(1997).

[7] **F. Herrera, M.Lozano, J.L.Verdegay**: A learning process for fuzzy control rules using genetic algorithms, Fuzzy Sets and Systems, Vol.100, N.1-3(1998).

[8] **C. Li and R. Priemer**: Fuzzy control of unknown multiple-input-multiple-output plants, Fuzzy Sets and Systems, Vol.104, N.2(1999).

[9] **M.R. Emani, I.B. Turksen, A.A. Goldenberg**: A unified parameterized formulation of reasoning in fuzzy modeling and control, Fuzzy Sets and Systems, Vol.108, N.1(1999).

6.9 Lý thuyết tập hợp

[1] **E.Tresch**: On the convergence of product- sum series of L- R fuzzy numbers, Fuzzy Sets and Systems, Vol.53, N.2(1993).

[2] **X. Xiaoping, H. Minghu and W. Congxin**: On the extension of the fuzzy number measure in Banach spaces: Part I. Representation of fuzzy number measure, Fuzzy Sets and Systems. Vol.78, N.3(1996).

[3] **D.P. Filev, R.R. Yager**: Operations of fuzzy numbers via fuzzy reasoning. Fuzzy Sets and Systems, Vol.91, N.2(1997).

[4] **K.I. Zhang, K. Hirota**: On fuzzy number lattice (R1 ≤,)Fuzzy Sets and Systems, Vol.92, N.1(1997).

[5] **M. Delgado, M.A. Vila, W. Voxman**: On a canonical representation of fuzzy numbers, Fuzzy Sets and Systems, Vol.93, N.1(1998).

[6] **W.Voxman:** Some remarks on distances between fuzzy numbers, Fuzzy Sets and Systems, Vol.100, N.1-3(1998).

[7] **D. Dubois and H. Prade:** Semantics of quotient operators in fuzzy relations databases, Fuzzy Sets and Systems, Vol.78, N.1(1996).

[8] **F. Tamaki, A. Kanagawa, H. Ohta:** Identification of membership functions based on fuzzy observation data, Fuzzy Sets and Systems, Vol. 93, N.3(1998).

3 LOGIC MỜ VÀ CÁC ỨNG DỤNG ĐA DẠNG CỦA NÓ

Bùi Công Cường
Viện Toán học Hà nội

Tóm tắt

Bài tổng quan này gồm 2 phần. Phần 1 sẽ trình bày một số *kiến thức cơ bản của logic mờ* - một hướng hiện đại, đã và đang được khẩn trương nghiên cứu trong nước cũng như trên thế giới và thực tiễn đã tìm được rất nhiều ứng dụng trong nhiều lĩnh vực khác nhau: sản xuất công nghiệp, công nghệ tin học, nghiên cứu và dự báo kinh tế, quản lý môi trường, xã hội học, y học, toán học, ...

Phần 2 của bài báo tập trung vào giới thiệu nhanh một số ứng dụng quan trọng nhất của công nghệ mờ trong các lĩnh vực: điều khiển mờ, hệ chuyên gia mờ, nhận dạng mờ, hệ hỗ trợ quyết định và bài toán lấy quyết định.

Vào đầu

Bất kỳ một người nào có ít nhiều tri thức đều hiểu rằng ngay trong những suy luận đời thường cũng như trong các suy luận khoa học chặt chẽ,hay khi triển khai ứng dụng, logic toán học cổ điển và nhiều định lý toán học quan trọng thu được qua những lập luận bằng logic cổ điển đã đóng vai trò rất quan trọng.

Nhưng đáng tiếc, chiếc áo logic toán học cổ điển đã quá chật hẹp đối với những ai mong muốn tìm kiếm những cơ sở vững chắc cho những suy luận phù hợp hơn với những bài toán nảy sinh từ công việc nghiên cứu và thiết kế những hệ thống phức tạp, đặc biệt là những cố gắng đưa những suy luận giống như cách con người vẫn thường sử dụng vào các lĩnh vực trí tuệ nhân tạo (chẳng hạn, như trong các hệ chuyên gia, các hệ hỗ trợ quyết định, ...) hay vào trong công việc điều khiển và vận hành các hệ thống lớn, phức tạp sao cho kịp thời và hiệu quả.

Một cách tiếp cận mới đã mang lại nhiều kết quả thực tiễn và có nhiều triển vọng tiếp tục phát triển mạnh mẽ. Đó là cách tiếp cận của *lý thuyết tập mờ* (Fuzzy Set Theory), bắt đầu với công trình của L. Zadeh, 1965 [1]. Trong sự phát triển đa dạng của lý thuyết tập mờ và các hệ mờ, *logic mờ* (Fuzzy Logic) giữ một vai trò cơ bản. Trong bài tổng quan gọn này chúng tôi sẽ trình bày một số kiến thức cơ bản của logic mờ thông qua việc giới thiệu các phép toán chủ chốt, vấn đề lập luận xấp xỉ (lập luận mờ) của logic mờ bằng con đường đủ hiện đại và trực quan và một số ứng dụng của nó.

Phần I bài viết sẽ tập trung vào những phép toán cơ bản và bước đầu đi vào lập luận xấp xỉ với phép suy diễn mờ. Một vài tính toán trong phần cuối nhằm minh họa cho điều khẳng

định của người viết là logic mờ tuy còn để ngỏ nhiều bài toán, song những bước ứng dụng ban đầu vào các bài toán đơn giản không phải là quá khó khăn.

Khả năng ứng dụng rất đa dạng của logic mờ sẽ tìm thấy trong phần II của bài viết và trong các tài liệu tìm thấy trong phần tài liệu trích dẫn (phần lớn là những tài liệu có thể tìm được ngay trong nước).

1 Kiến thức cơ bản về logic mờ

1.1 Ôn nhanh về logic mệnh đề cổ điển

Ta sẽ kí hiệu $\mathcal{P}$ là tập hợp các mệnh đề và $P, P_1, Q, Q_1, \ldots$ là những mệnh đề. Với mỗi mệnh $P \in \mathcal{P}$, ta gán một trị $v(P)$ là giá trị chân lý (truth value) của mệnh đề. Logic cổ điển đề nghị $v(P)=1$, nếu P là đúng (T-true), $v(P)=0$, nếu P là sai (F-false).

Trên $\mathcal{P}$ chúng ta xác định trước tiên 3 phép toán cơ bản và rất trực quan:

Phép tuyển P OR Q, kí hiệu $P \vee Q$, đó là mệnh đề "hoặc P hoặc Q".

Phép hội P AND Q, kí hiệu $P \wedge Q$, đó là mệnh đề "vừa P vừa Q".

Phép phủ định NOT P, kí hiệu $\rceil P$, đó là mệnh đề " không P ".

Dựa vào 3 phép toán logic cơ bản này người ta đã định nghĩa nhiều phép toán khác, nhưng đối với chúng ta quan trọng nhất là phép kéo theo (*implication*), sẽ kí hiệu là $P \Rightarrow Q$.

Sử dụng những định nghĩa trên, trong logic cổ điển, các luật suy diễn quan trọng sau đây giữ vai trò rất quyết định trong các lập luận truyền thống. Đó là các luật

a) modus ponens: $(P \wedge (P \Rightarrow Q)) \Rightarrow Q$
b) modus tollens: $((P \Rightarrow Q) \wedge \rceil Q) \Rightarrow \rceil P$
c) syllogism: $((P \Rightarrow Q) \wedge (Q \Rightarrow R)) \Rightarrow (P \Rightarrow R)$
d) contraposition: $(P \Rightarrow Q) \Rightarrow (\rceil Q \Rightarrow \rceil P)$.

Ta hãy lấy luật modus ponens làm ví dụ. Luật này có thể giải thích như sau: "Nếu mệnh đề P là đúng và nếu định lý "P kéo theo Q " đúng, thì mệnh đề Q cũng đúng".

1.2 Logic mờ

1973 L.Zadeh trong tài liệu [4] đã đưa vào khái niệm "*biến ngôn ngữ* " và bước đầu ứng dụng vào suy diễn mờ – phần cơ bản của logic mờ. Đây là bước khởi đầu rất quan trọng cho công việc tính toán các suy diễn chủ chốt trong các hệ mờ.

Để có thể tiến hành mô hình hoá các hệ thống và biểu diễn các quy luật vận hành trong các hệ thống này, trước tiên chúng ta cần tới suy rộng các phép toán logic cơ bản (logic connectives) với các mệnh đề có giá trị chân lý $v(P)$ trong đoạn [0,1] (thay cho quy định $v(P)$ chỉ nhận giá trị 1 hoặc 0 như trước đây).

Chúng ta sẽ đưa vào các phép toán cơ bản của logic mờ qua con đường tiên đề hoá. Như vậy có lẽ tự nhiên và phần nào hứa hẹn sẽ có tính công nghệ hơn.

Cho các mệnh đề P, Q, P_1, ... , giá trị chân lý $v(P)$, $v(Q)$, $v(P_1)$, ... sẽ nhận trong đoạn [0,1]. Sau đây chúng ta đi ngay vào các phép toán cơ bản nhất.

1.2.1 Phép phủ định

Phủ định (negation) là một trong những phép toán logic cơ bản. Để suy rộng chúng ta cần tới toán tử v(NOT P) xác định giá trị chân lý của NOT P đối với mỗi mệnh đề NOT P.

Ta sẽ xét tới một số tiên đề diễn đạt những tính chất quen biết nhất vẫn dùng trong logic cổ điển:

e) v(NOT P) chỉ phụ thuộc vào $v(P)$.

f) Nếu $v(P) = 1$, thì v(NOT P) = 0.

g) Nếu $v(P) = 0$, thì v(NOT P) = 1.

h) Nếu $v(P_1) \leq v(P_2)$, thì v(NOT P_1) $\geq$ v(NOT P_2).

Bây giờ chúng ta cho dạng toán học của những toán tử này.

Định nghĩa 1: Hàm $n:[0,1] \to [0,1]$ không tăng thoả mãn các điều kiện $n(0)=1$, $n(1)=0$, gọi là *hàm phủ định* (negation – hay là phép phủ định).

Chúng ta có thể xét thêm vài tiên đề khác:

a) Nếu $v(P_1) < v(P_2)$, thì v(NOT P_1) > v(NOT P_2).

b) v(NOT P) phụ thuộc liên tục vào $v(P)$.

c) v(NOT(NOT P)) = $v(P)$.

Định nghĩa 2: 1) Hàm phủ định n là *chặt* (strict) nếu nó là hàm liên tục và giảm chặt.

2) Hàm phủ định n là *mạnh* (strong) nếu nó giảm chặt và thoả mãn: $n(n(x)) = x$ với mỗi x.

Vài ví dụ:

– Hàm phủ định chuẩn $n(x) = 1 - x$ (ví dụ trong định nghĩa của Zadeh [1]).

– Hàm phủ định $n(x) = 1 - x^2$.

– Phủ định trực cảm (Yage, 1980) $n(x) = 1$, nếu $x = 0$ và $n(x) = 0$ nếu $x > 0$.

– Họ phủ định (Sugeno, 1977) $N_\lambda(x) = \dfrac{1-x}{1+\lambda x}$, với $\lambda > -1$.

Định lý biểu diễn.

Xem trong [25].

1.2.3 Một cách định nghĩa phần bù của một tập mờ

Cho Ω là *không gian nền*, một *tập mờ* A trên Ω tương ứng với một hàm thực nhận giá trị trong đoạn $[0,1]$:

$A: \Omega \to [0,1]$, là hàm thuộc (*membership function*).

Người ta cũng dùng kí hiệu hàm thuộc $\mu_A: \Omega \to [0,1]$.

Chúng ta kí hiệu

$A = \{(a, \mu_A(a)): a \in \Omega\}$.

ở đây

$A(a) = \mu_A(a) \in [0,1]$.

là *độ thuộc* (membership degree) của phần tử x vào tập mờ A. Kí hiệu $\mu_A(a)$ hay được dùng hơn trong các tài liệu về mờ. Song vì thuận lợi chúng ta sẽ dùng $A(a)$.

Định nghĩa 3: Cho n là hàm phủ định, *phần bù* A^C của tập mờ A là một tập mờ với hàm thuộc được xác định bởi $A^C(a) = n(A(a))$, với mỗi $a \in \Omega$.

1.2.4 Phép hội

Phép hội (vẫn quen gọi là phép AND – *conjunction*) là một trong mấy phép toán logic cơ bản nhất. Nó cũng là cơ sở để định nghĩa phép giao của hai tập mờ. Chúng ta cần xem xét các tiên đề sau:

a) $v(P_1 \text{ AND } P_2)$ chỉ phụ thuộc vào $v(P_1)$, $v(P_2)$.

b) Nếu $v(P_1)=1$, thì $v(P_1 \text{ AND } P_2) = v(P_2)$, với mọi mệnh đề P_2.

c) Giao hoán: $v(P_1 \text{ AND } P_2) = v(P_2 \text{ AND } P_1)$.

d) Nếu $v(P_1) \le v(P_2)$ thì $v(P_1 \text{ AND } P_3) \le v(P_2 \text{ AND } P_3)$, với mọi mệnh đề P_3.

e) Kết hợp: $v(P_1 \text{ AND } (P_2 \text{ AND } P_3)) = v((P_1 \text{ AND } P_2) \text{ AND } P_3)$.

Nếu diễn đạt *phép hội mờ* (fuzzy conjunction) như một hàm $T:[0,1]^2 \to [0,1]$ thì chúng ta có thể cần tới các hàm sau:

Định nghĩa 4: Hàm $T:[0,1]^2 \to [0,1]$ là một *t–chuẩn* (*chuẩn tam giác* hay *t–norm*) khi và chỉ khi thoả mãn các điều kiện sau:

a) $T(1,x) = x$, với mọi $0 \le x \le 1$,

b) T có tính giao hoán, tức là $T(x,y) = T(y,x)$, với mọi $0 \le x, y \le 1$,

c) T không giảm theo nghĩa $T(x,y) \le T(u,v)$, với mọi $x \le u$, $y \le v$,

d) T có tính kết hợp: $T(x, T(y,z)) = T(T(x,y), z)$ với mọi $0 \le x, y, z \le 1$.

Từ những tiên đề trên chúng ta suy ra ngay $T(0,x)$. Hơn nữa tiên đề d) đảm bảo tính thác triển duy nhất cho hàm nhiều biến.

Định lý biểu diễn: Xem trong [25].

Một vài lớp t–chuẩn khác cùng một số tính chất các bạn có thể tìm thấy trong bài báo của Klement [6]và E. Walker [23, trang 306-314].

1.2.5 Định nghĩa tổng quát phép giao của hai tập mờ.

Cho hai tập mờ A, B trên cùng không gian nền Ω với hàm thuộc $A(a), B(a)$. Cho T là một t–chuẩn.

Định nghĩa 5: Ứng với t–chuẩn T, *tập giao* (tổng quát) của hai tập mờ A, B là một tập mờ $(A \cap_T B)$ trên Ω với hàm thuộc cho bởi:

$(A \cap_T B)(a) = T(A(a), B(a))$, với mọi $a \in \Omega$.

Việc lựa chọn phép giao nào, tức là chọn t–chuẩn T nào để làm việc và tính toán hoàn toàn phụ thuộc vào từng bài toán cụ thể mà bạn đang quan tâm.

Ví dụ: Hamacher 1978 đề nghị dùng

$$(A \cap_T B)(a) = \frac{A(a)B(b)}{p + (1-p)[A(a) + B(b) - A(a)B(a)]}, \ p \geq 0, \text{ với mỗi } a \in \Omega.$$

còn Yager,1980 xét phép giao hai tập mờ A, B với hàm thuộc cho bởi

$$(A \cap_T B)(a) = 1 - \min\{1, ((1-A(a))^p + (1-B(a))^p)^{1/p}\}, \ p \geq 1, \text{ với mỗi } a \in [0,1].$$

Cùng thời, Dubois và Prade cũng đề nghị một họ toán tử phụ thuộc tham số t, đó là phép giao $(A \cap_T B)$ với hàm thuộc

$$(A \cap_T B)(a) = \frac{A(a)B(a)}{\max\{A(a), B(a), t\}}, \text{ với } 0 \leq t \leq 1, \text{ và mọi } a \in [0,1].$$

1.2.6 Phép tuyển

Giống như phép hội, phép tuyển hay toán tử logic OR (*disjunction*) thông thường cần thoả mãn các tiên đề sau:

a) $v(P_1 \text{ OR } P_2)$ chỉ phụ thuộc vào $v(P_1)$, $v(P_2)$.

b) Nếu $v(P_1)=0$, thì $v(P_1 \text{ OR } P_2) = v(P_2)$, với mọi mệnh đề P_2.

c) Giao hoán: $v(P_1 \text{ OR } P_2) = v(P_2 \text{ OR } P_1)$.

d) Nếu $v(P_1) \leq v(P_2)$ thì $v(P_1 \text{ OR } P_3) \leq v(P_2 \text{ OR } P_3)$, với mọi mệnh đề P_3.

e) Kết hợp: $v(P_1 \text{ OR } (P_2 \text{ OR } P_3)) = v((P_1 \text{ OR } P_2) \text{ OR } P_3)$.

Khi ấy chúng ta có thể nghĩ tới các phép tuyển được định nghĩa bằng con đường tiên đề như sau:

Định nghĩa 6: Hàm $S : [0.1]^2 \rightarrow [0.1]$ gọi là *phép tuyển* (OR suy rộng) hay là *t–đối chuẩn* (*t–conorm*) nếu thoả mãn các tiên đề sau:

a) $S(0.x) = x$ với mọi $x \in [0,1]$.

b) S có tính giao hoán: $S(x,y) = S(y,x)$ với mọi $0 \le x,y) \le 1$.

c) S không giảm: $S(x,y) \le S(u,v)$ với mọi $0 \le x \le u \le 1$ và $0 \le y \le v \le 1$

d) S có tính kết hợp: $S(x,S(y,z)) = S(S(x,y),z)$ với mọi $0 \le x,y,z \le 1$

Định lý 7: Cho n là phép phủ định mạnh. T là một t–chuẩn. khi ấy hàm S xác định trên $[0,1]^2$ bằng biểu thức

$S(x,y) = n T(nx, ny)$ với mọi $0 \le x,y \le 1$,

là một t–đối chuẩn.

Chứng minh: Xem [25].

Không khó khăn nhận thấy rằng. với t–đối chuẩn $S(x,y)$ bất kỳ thực hiện bất đẳng thức

$\max(x,y) \le S(x,y) \le Z'(x,y)$.

Định lý biểu diễn: Xem trong [25] và E. Walker [23, trang 306-314].

Định lý 8: Cho S là t–đối chuẩn. Khi ấy:

a) S gọi là liên tục nếu đó là hàm liên tục trên miền xác định.

b) S là Archimed nếu $S(x,x) > x$. với mỗi $0<x<1$.

c) S gọi là chặt nếu S là hàm tăng tại mỗi điểm $(x,y) \in (0,1)^2$.

1.2.7 Định nghĩa tổng quát phép hợp của hai tập mờ

Định nghĩa 9: Cho Ω là không gian nền. A,B là hai tập mờ trên Ω với hàm thuộc $A(a),B(a)$. S là t–đối chuẩn. Phép hợp $(A \cup_S B)$ trên Ω của hai tập mờ là một tập mờ với hàm thuộc:

$(A \cup_S B)(a) = S(A(a),B(a))$, với mọi $a \in \Omega$.

Việc lựa chọn phép hợp nào, tức là chọn t–đối chuẩn S nào để xác định hàm thuộc tương ứng phụ thuộc vào bài toán đang nghiên cứu. Sau đây là mấy ví dụ:

– Hamacher .1978 . đã cho họ phép hợp hai tập mờ với hàm thuộc theo tham số q:

$$(A \cup_S B)(a) = \frac{(q-1)A(a)B(a) + A(a) + B(a)}{1 + qA(a)B(a)}, q \ge -1, \text{ với } a \in \Omega .$$

– Còn họ phép hợp $(A \cup_S B)$ tương ứng của Yager cho bởi hàm thuộc với tham số q:

$$(A \cup_S B)(a) = \min\{1, (A(a)^p + B(a)^p)^{1/p}\}, \text{ với } p \ge 1. \text{ với } a \in \Omega .$$

– Tương tự. họ phép hợp do Dubois và Prade đề nghị với các hàm thuộc với tham số t. có dạng:

$$(A \cup_S B)(a) = \frac{A(a) + B(a) - A(a)B(a) - \min\{A(a), B(a), (1-t)\}}{\max\{(1 - A(a)), (1 - B(a)), t\}}$$, với $t \in [0,1]$, $a \in \Omega$.

1.2.8 Một số quy tắc với phép hội và phép tuyển

Nhiều bạn đọc trong nghiên cứu hay chứng minh thường quen dùng nhiều quy tắc suy luận (hay đơn giản hơn là sử dụng một số tính chất gần như hiển nhiên), song thực ra những quy tắc đó có được là do chúng ta xây phần toán học trước đây trên lý thuyết tập hợp cổ điển và logic cổ điển. Chuyển sang lý thuyết tập mờ và suy luận với logic mờ chúng ta cần thận trọng với những thói quen cũ này.

Ví dụ trong lý thuyết tập hợp, với bất kỳ tập rõ $A \subset \Omega$ thì

$A \cap A^C = \varnothing$, $A \cup A^C = \Omega$.

nhưng sang tập mờ thì hai tính chất quen dùng đó không còn đúng nữa.

Sau đây chúng ta dừng lại với mấy quy tắc quen biết của hai phép toán hội và phép tuyển.

Cho T là một t–chuẩn, S là t–đối chuẩn.

Tính luỹ đẳng

Định nghĩa 10: Chúng ta nói T là luỹ đẳng (*idempotency*) nếu $T(x,x)=x$, với mọi $x \in [0,1]$, S là luỹ đẳng nếu $S(x,x)=x$, với mọi $x \in [0,1]$.

Mệnh đề 11: T là luỹ đẳng khi và chỉ khi $T(x,y)=\min(x,y)$, với $x,y \in [0,1]$,
S là luỹ đẳng khi và chỉ khi $S(x,y)=\max(x,y)$, với $x,y \in [0,1]$.

Tính hấp thụ

Định nghĩa 12: Có hai dạng định nghĩa hấp thụ (*absorption*) suy rộng từ lý thuyết tập hợp:

a) $T(S(x,y),x)= x$ với mọi $x,y \in [0,1]$. (1)

b) $S(T(x,y),x)= x$ với mọi $x,y \in [0,1]$. (2)

Mệnh đề 13: Đẳng thức (1) thực hiện khi và chỉ khi $T(x,y)=\min(x,y)$, $\forall x,y \in [0,1]$.
Đẳng thức (2) thực hiện khi và chỉ khi $S(x,y)=\max(x,y)$, $\forall x,y \in [0,1]$.

Tính phân phối

Định nghĩa 14: Có hai biểu thức xác định tính phân phối (distributivity):

a) $S(x,T(y,z))=T(S(x,y),S(x,z))$, với mọi $x,y,z \in [0,1]$. (3)

b) $T(x,S(y,z))=S(T(x,y),T(x,z))$, với mọi $x,y,z \in [0,1]$. (4)

Mệnh đề 15: Đẳng thức (3) thực hiện khi và chỉ khi $T(x,y)=\min(x,y)$, $\forall x,y \in [0,1]$.
Đẳng thức (4) thực hiện khi và chỉ khi $S(x,y)=\max(x,y)$, $\forall x,y \in [0,1]$.

Như vậy nhiều tính chất quen biết hay dùng chỉ luôn luôn đúng với hai phép toán min và max.

1.2.9 Luật De Morgan

Trong lý thuyết tập hợp luật De Morgan nổi tiếng sau đây được sử dụng nhiều nơi: Cho A, B là hai tập con của Ω, khi đó

$$(A \cup B)^C = A^C \cap B^C$$

và $$(A \cap B)^C = A^C \cup B^C$$

Có nhiều dạng suy rộng hai đẳng thức này. Sau đây một dạng suy rộng cho logic mờ.

Định nghĩa 16: Cho T là t–chuẩn, S là t–đối chuẩn, n là phép phủ định chặt. Chúng ta nói bộ ba (T, S, n) là một *bộ ba De Morgan* nếu

$$n(S(x, y)) = T(nx, ny).$$

Chúng ta nói bộ ba (T, S, n) là liên tục nếu T và S là hai hàm liên tục. Sau đây là 2 lớp bộ ba quan trọng:

Định nghĩa 17: Bộ ba De Morgan (T, S, n) là bộ ba *mạnh* (strong) khi và chỉ khi có một tự đồng cấu $\varphi : [0,1] \to [0,1]$ sao cho:

a) $T(x, y) = \varphi^{-1}(\max\{\varphi(x) + \varphi(y) - 1 \ , \ 0\})$.

b) $S(x, y) = \varphi^{-1}(\min\{\varphi(x) + \varphi(y) \ , \ 1\})$.

c) $N(x) = \varphi^{-1}(1 - \varphi(x))$.

Định nghĩa 18: Bộ ba De Morgan (T, S, n) là bộ ba *chặt* (strict) khi và chỉ khi có một tự đồng cấu $\varphi : [0,1] \to [0,1]$ sao cho :

a) $T(x, y) = \varphi^{-1}(\varphi(x)\varphi(y))$.

b) $S(x, y) = \varphi^{-1}(\varphi(x) + \varphi(y) - \varphi(x)\varphi(y))$.

c) $N(x) = \varphi^{-1}(1 - \varphi(x))$.

1.2.10 Phép kéo theo

Cho đến bây giờ đã có khá nhiều nghiên cứu về phép kéo theo (implication). Điều đó cũng tự nhiên vì đây là công đoạn chốt nhất của quá trình suy diễn trong mọi lập luận xấp xỉ ,bao gồm cả suy luận mờ. Trong phần tiếp theo này chúng ta sẽ đi tiếp con đường tiên đề hoá và sau đó dừng nhanh tại vài dạng phổ cập để minh họa.

Chúng ta sẽ xét phép kéo theo như một mối quan hệ, một toán tử logic. Thông thường chúng ta nhớ tới các tiên đề sau cho hàm $v(P_1 \Rightarrow P_2)$.:

1) $v(P_1 \Rightarrow P_2)$ chỉ phụ thuộc vào giá trị $v(P_1)$, $v(P_2)$.
2) Nếu $v(P_1) \le v(P_3)$ thì $v(P_1 \Rightarrow P_2) \ge v(P_3 \Rightarrow P_2)$, với mọi mệnh đề P_2.
3) Nếu $v(P_2) \le v(P_3)$ thì $v(P_1 \Rightarrow P_2) \le v(P_1 \Rightarrow P_3)$, với mọi mệnh đề P_1.
4) Nếu $v(P_1) = 0$ thì $v(P_1 \Rightarrow P) = 1$, với mỗi mệnh đề P.
5) Nếu $v(P_1) = 1$ thì $v(P \Rightarrow P_1) = 1$, với mỗi mệnh đề P.

6) Nếu $v(P_1) = 1$ và $v(P_2) = 0$, thì $v(P_1 \Rightarrow P_2) = 0$.

Tính hợp lý của những tiên đề này chủ yếu dựa vào logic cổ điển và những tư duy trực quan về phép suy diễn. Từ tiên đề I0 chúng ta khẳng định sự tồn tại hàm số $I(x,y)$ xác định trên $[0,1]^2$ với mong muốn đo giá trị chân lý của phép kéo theo qua biểu thức

$$v(P_1 \Rightarrow P_2) = I(v(P_1), v(P_2)).$$

Định nghĩa 19: *Phép kéo theo* (implication) là một hàm số $I:[0,1]^2 \to [0,1]$ thỏa mãn các điều kiện sau:

1) Nếu $x \le z$ thì $I(x,y) \ge I(z,y)$ với mọi $y \in [0,1]$,
2) Nếu $y \le u$ thì $I(x,y) \le I(x,u)$ với mọi $x \in [0,1]$,
3) $I(0,x) = 1$ với mọi $x \in [0,1]$,
4) $I(x,1) = 1$ với mọi $x \in [0,1]$,
5) $I(1,0) = 0$.

Để ý rằng tuy rất đơn giản nhưng điều kiện 5) vẫn cần đưa vào định nghĩa vì không thể suy ra từ 4 tiên đề trên.

Từ định nghĩa toán học dễ dàng nhận thấy mỗi phép kéo theo là một tập mờ trên $[0,1]^2$ và như vậy xác lập một quan hệ mờ trên $[0,1]^2$.

Tiếp tục, chúng ta xem xét thêm một số tính chất khác của phép kéo theo, những tính chất này nhận được nhờ những bài báo của Dubois và Prade.

1) $I(1,x)=x$, với mọi $x \in [0,1]$.
2) $I(x,I(y,z))=I(y,I(x,z))$.

   Đây là quy tắc đổi chỗ, cơ sở trên sự tương đương giữa hai mệnh đề:

   "If P_1 then (If P_2 then P_3)"

   và "If (P_1 AND P_2) then P_3)".

3) $x \le y$ nếu và chỉ nếu $I(x,y)=1$.

   Tiên đề 3) này biểu thị ý: phép kéo theo xác lập một thứ tự.

4) $I(x,0))=N(x)$ là một phép phủ định mạnh, như vậy 4) phản ánh mệnh đề sau từ logic cổ điển $P \Rightarrow Q = \rceil P$ nếu $v(Q)=0$ (nếu Q là False).
5) $I(x,y) \ge y$, với mọi x,y.
6) $I(x,x)=1$ với mọi x.
7) $I(x,y)=I(N(y),N(x))$. Điều kiện này phản ánh phép suy luận ngược trong logic cổ điển 2 giá trị: $(P \Rightarrow Q) = (\rceil Q \Rightarrow \rceil P)$. Nói chung đây là một điều kiện mạnh.
8) $I(x,y)$ là hàm liên tục trên $[0,1]^2$.

Để tìm hiểu thêm các điều kiện này chúng ta xét tới định lý sau.

Định lý 20: Mỗi hàm số $I:[0,1]^2 \to [0,1]$ thoả mãn các điều kiện 2), 7), 8) thì cũng sẽ thỏa mãn các điều kiện 1), 3), 4), 5), 6), 10) và 11).

Chứng minh: Xem trong [25]

1.2.11 Một số dạng hàm kéo theo cụ thể

Để tính toán được . chúng ta cần những dạng cụ thể của phép kéo theo. Sau đây là một số dạng hàm kéo theo, xây dựng dựa vào các phép toán logic mờ đã suy rộng phía trên.

Cho T là t–chuẩn, S là t–đối chuẩn, n là phép phủ định mạnh.

Định nghĩa 19: *Dạng kéo theo thứ nhất.* Hàm $I_{S1}(x,y)$ xác định trên $[0,1]^2$ bằng biểu thức

$I_{S1}(x,y) = S(n(x),y)$.

Rõ ràng ẩn ý sau định nghĩa này là công thức từ logic cổ điển $P \Rightarrow Q = \rceil P \vee Q$.

Định lý 20: Với bất kỳ t–chuẩn T, t–đối chuẩn S và phép phủ định mạnh n nào, I_{S1} là một phép kéo theo thỏa mãn định nghĩa 19.

Chứng minh: Xem trong [25]

Phép kéo thứ hai sau đây lấy ý từ logic trực cảm (intuitionistic logic).

Định nghĩa 21: Cho T là t-chuẩn, hàm $I_T(x,y)$ xác định trên $[0,1]^2$ bằng biểu thức

$I_T(x,y) = \sup\{u : T(x,u) \le y\}$.

Định lý 22: Với bất kỳ t–chuẩn T nào, I_T được định nghĩa như trên là một phép kéo theo thõa mãn định nghĩa 19.

Chứng minh: Xem trong [25]

1.3 Quan hệ mờ

1.3.1 Quan hệ mờ và phép hợp thành

Định nghĩa 23: Cho X, Y là hai không gian nền. R gọi là một *quan hệ mờ* trên $X \times Y$ nếu R là một tập mờ trên $X \times Y$, tức là có một hàm thuộc $\mu_R: X \times Y \to [0,1]$, ở đây $\mu_R(x,y) = R(x,y)$ là độ thuộc (*membership degree*) của (x,y) vào quan hệ R.

Định nghĩa 24: Cho R_1 và R_2 là hai quan hệ mờ trên $X \times Y$, ta có định nghĩa

a) Quan hệ $R_1 \cup R_2$ với $\mu_{R_1 \cup R_2}(x,y) = \max\{\mu_{R_1}(x,y), \mu_{R_2}(x,y)\}$, $\forall (x,y) \in X \times Y$.

b) Quan hệ $R_1 \cap R_2$ với $\mu_{R_1 \cap R_2}(x,y) = \min\{\mu_{R_1}(x,y), \mu_{R_2}(x,y)\}$, $\forall (x,y) \in X \times Y$.

Định nghĩa 25: *Quan hệ mờ trên những tập mờ.* Cho tập mờ A với $\mu_A(x)$ trên X, tập mờ B với $\mu_B(x)$ trên Y. Quan hệ mờ trên các tập mờ A và B là quan hệ mờ R trên $X \times Y$ thoả mãn điều kiện:

$\mu_R(x,y) \le \mu_A(x)$, $\forall y \in Y$ và $\mu_R(x,y) \le \mu_B(x)$, $\forall x \in X$.

Định nghĩa 26: Cho quan hệ mờ R trên $X\times Y$.

Phép chiếu của R lên X là: $\text{proj}_X R = \{(x, \max_y \mu_R(x,y)) : x\in X\}$

Phép chiếu của R lên Y là: $\text{proj}_Y R = \{(y, \max_x \mu_R(x,y)) : y\in Y\}$

Định nghĩa 27: Cho quan hệ mờ R trên $X\times Y$. Thác triển R lên không gian tích $X\times Y\times Z$ là:

$\text{ext}_{XYZ} R = \{(x,y,z), \mu_{\text{ext}}(x,y,z) = \mu_R(x,y)\ , \forall z\in Z\}$.

1.3.2 Phép hợp thành

Định nghĩa 28: Cho R_1 là quan hệ mờ trên $X\times Y$ và R_2 là quan hệ mờ trên $Y\times Z$. Hợp thành $R_1\circ R_2$ của R_1, R_2 là quan hệ mờ trên $X\times Z$.

a) *Hợp thành max-min* (max–min composition) được xác định bởi

$$\mu_{R_1\circ R_2}(x,z) = \max_y\{\min(\mu_{R_1}(x,y), \mu_{R_2}(y,z)\},\ \forall(x,z)\in X\times Z.$$

b) *Hợp thành max-prod* cho bởi

$$\mu_{R_1\circ R_2}(x,z) = \max_y\{(\mu_{R_1}(x,y)\cdot\mu_{R_2}(y,z)\},\ \forall(x,z)\in X\times Z.$$

c) *Hợp thành max–** được xác định bởi toán tử $*: [0.1]^2 \to [0.1]$

$$\mu_{R_1\circ R_2}(x,z) = \max_y\{\mu_{R_1}(x,y) * \mu_{R_2}(y,z)\},\ \forall(x,z)\in X\times Z.$$

Giả thiết (T,S,n) là bộ ba De Morgan, trong đó T là t-chuẩn, S là t-đối chuẩn, n là phép phủ định.

Định nghĩa 29: Cho R_1, R_2 là quan hệ mờ trên $X\times X$, phép T–tích hợp thành cho một quan hệ $R_1\circ_T R_2$ trên $X\times X$ xác định bởi

$R_1\circ_T R_2(x,z) = \sup_{y\in X} T(R_1(x,y), R_2(y,z))$.

Định lý 30: Cho R_1, R_2, R_3 là những quan hệ mờ trên $X\times X$, khi đó:

a) $R_1\circ_T(R_2\circ_T R_3) = (R_1\circ_T R_2)\circ_T R_3$.

b) Nếu $R_1\subseteq R_2$ thì $R_1\circ_T R_3 \subseteq R_2\circ_T R_3$ và $R_3\circ_T R_1 \subseteq R_3\circ_T R_2$.

1.3.3 Tính chuyển tiếp

Định nghĩa 31: Quan hệ mờ R trên $X\times X$ gọi là:

a) *min–chuyển tiếp* nếu $\min\{R(x,y), R(y,z)\} \le R(x,z)\ \ \forall x,y,z\in X$..

b) *chuyển tiếp yếu* nếu $\forall x,y,z\in X$ có

$$R(x,y) > R(y,x) \text{ và } R(y,z) > R(z,y) \text{ thì } R(x,z) > R(z,x).$$

c) *chuyển tiếp tham số* nếu có một số $0<\theta<1$ sao cho: Nếu $R(x,y) > \theta > R(y,x)$ và $R(y,z) > \theta > R(z,y)$ thì $R(x,z) > \theta > R(z,x)\ \ \forall x,y,z\in X$.

Định lý 32:

a) Nếu R là quan hệ mờ có tính chất *min–chuyển tiếp* thì R là quan hệ mờ có tính chất *chuyển tiếp tham số* với mọi $0< \theta < 1$.

b) Nếu R là quan hệ mờ có tính chất *chuyển tiếp tham số* thì R là quan hệ mờ có tính chất *chuyển tiếp yếu*.

Chứng minh: Xem trong [24]

1.3.4 Phương trình quan hệ mờ

Phương trình quan hệ mờ lần đầu tiên nghiên cứu bởi GS. Sanchez năm 1976, đóng vai trò quan trọng trong các lĩnh vực phân tích các hệ mờ, thiết kế các bộ điều khiển mờ, quá trình lấy quyết định và nhận dạng mờ.

Dạng đơn giản nhất có thể diễn đạt như sau:

Cho một hệ mờ biểu diễn dưới dạng một quan hệ mờ nhị nguyên R trên không gian tích $X \times Y$. Đầu vào (input) của hệ là một tập mờ A cho trên không gian nền input X. Tác động của đầu vào A với hệ R sẽ là phép hợp thành $A \circ R$ sẽ cho ở đầu ra (output) một tập mờ trên không gian nền Y, kí hiệu là B. Khi ấy chúng ta có $A \circ R = B$.

Nếu chúng ta sử dụng phép hợp thành max–min thì hàm thuộc của B cho bởi

$$\mu_B(y = \mu_{A \circ R}(y) = \max_x(\min_y[\mu_A(x), \mu_R(x, y)])$$

Ví dụ 33: Cho input là tập mờ A trên X và quan hệ mờ R trên $X \times Y$ như sau:

$$X=\{x_1, x_2, x_3\},\ Y=\{y_1, y_2, y_3\},\ A = \frac{0{,}2}{x_1} + \frac{0{,}8}{x_2} + \frac{1}{x_3} = \begin{pmatrix} 0.2 & 0.8 & 1 \end{pmatrix}$$

$$A \circ R = \begin{bmatrix} 0.7 & 1 & 0.4 \\ 0.5 & 0.9 & 0.6 \\ 0.2 & 0.6 & 0.3 \end{bmatrix},$$

Khi đó chúng ta có

$$B = A \circ R = \begin{pmatrix} 0{,}2 & 0{,}8 & 1 \end{pmatrix} \circ \begin{bmatrix} 0.7 & 1 & 0.4 \\ 0.5 & 0.9 & 0.6 \\ 0.2 & 0.6 & 0.3 \end{bmatrix} = \begin{pmatrix} 0{,}5 & 0{,}8 & 0{,}6 \end{pmatrix} = \frac{0{,}5}{y_1} + \frac{0{,}8}{y_2} + \frac{0{,}6}{y_3}.$$

1.4 Suy luận xấp xỉ và suy diễn mờ

1.4.1 Chúng ta sẽ trình bày đủ đơn giản vấn đề suy luận xấp xỉ dưới dạng những mệnh đề với các biến ngôn ngữ như đời thường vẫn dùng như: "máy lạnh", "ga yếu", hay những quy tắc, những luật dạng mệnh đề "nếu quay tay ga mạnh thì tốc độ xe sẽ nhanh".

Suy luận xấp xỉ - hay còn gọi là suy luận mờ - đó là quá trình suy ra những kết luận dưới dạng các mệnh đề mờ trong điều kiện các quy tắc, các luật, các dữ liệu đầu vào cho

trước cũng không hoàn toàn xác định. Chúng ta sẽ hạn chế bởi những luật đơn giản như dạng *modus ponens* hay *modus tollens* đã nêu ở phần đầu.

Trước tiên chúng ta nhớ lại trong giải tích toán học đã dùng quá trình lập luận sau:

Định lý:	Nếu một hàm số là khả vi thì nó liên tục
Sự kiện:	Hàm *f* khả vi
Kết luận:	*f* liên tục

đây là dạng suy luận dựa vào luật *modus ponens*. Bây giờ ta tìm cách diễn đạt cách suy luận quen thuộc trên dưới dạng sao cho có thể suy rộng cho logic mờ.

Ký hiệu: U = không gian nền = không gian tất cả các hàm số.

Ví dụ đơn giản có thể hiểu

$U = \{g : R \to R\}$.

A = {các hàm khả vi}.

B = {các hàm liên tục}.

Hãy chọn hai mệnh đề P = "$g \in A$" và Q = "$g \in B$". Khi ấy chúng ta có

Luật (tri thức):	$g \Rightarrow B$
Sự kiện:	P đúng (true)
Kết luận:	Q đúng (true)

ở đây chúng ta đã sử dụng luật *modus ponens* $((P \Rightarrow Q) \wedge P) \Rightarrow Q$.

1.4.2 Bây giờ đã có thể chuyển sang suy diễn mờ cùng dạng.

Luật mờ:	Nếu góc tay quay ga lớn thì xe đi nhanh
Sự kiện mờ:	Góc tay ga quay khá lớn
Hệ quả:	Xe đi khá nhanh

Zadeh đã diễn đạt sự kiện trên bằng các biến ngôn ngữ: góc tay quay, tốc độ, nhiệt độ, áp lực, tuổi tác và các mệnh đề mờ dạng tương ứng. Chúng ta làm rõ cách tiếp cận của Zadeh qua vài ví dụ.

1.4.2.1 Biến ngôn ngữ

Ví dụ 1: Ta nói "*Nam có tuổi trung niên*". khi ấy chọn

x = biến ngôn ngữ "*Tuổi*",

không gian nền là thời gian sống

U = [0, 130 năm].

A= tập mờ "*trung niên*".

Một cách tự nhiên, ta gán cho A là một tập mờ trên U với hàm thuộc $A(u) : U \to [0,1]$.

Sự kiện "*có thể tuổi của Nam là* 40" dĩ nhiên không chắc chắn và khá hợp lý nếu diễn đạt như một khả năng, trong [4,5] Zadeh đề nghị

Khả năng (Tuổi của Nam = 40) = Poss(x = 40)
= độ thuộc của số 40 vào tập mờ $A = A(40)$.

Mệnh đề mờ

"*Nam có tuổi trung niên*"

bây giờ được diễn đạt thành mệnh đề

$P = \{ x = A \}$ = {biến x nhận giá trị mờ A trên không gian nền U}
= {x is A } (theo dạng tiếng Anh).

1.4.2.2 Ví dụ 2: Đối với suy luận mờ cho ở đầu mục này chúng ta có thể dùng biến ngôn ngữ

x= "góc tay quay"

trên không gian nền U = [0.360°] (cho phép quay tay ga của xe máy), A = "góc lớn" là một tập mờ trên U (trong trường hợp này tiện hơn dùng khái niệm số mờ A), với hàm thuộc $A(u): U \rightarrow [0.1]$.

Tương tự, biến ngôn ngữ y = "tốc độ xe", với không gian nền

V = [0 km/giờ. 150 km/giờ].

Q = "xe đi nhanh"= một tập mờ B trên không gian nền V với hàm thuộc $B(v) : V \rightarrow [0,1]$. Khi ấy

P = "góc tay quay lớn" = $\{ x = A \}$ (x is u).

Q = "xe đi nhanh" = $\{ y = B \}$.

và luật mờ có dạng $P \Rightarrow Q$.

Như vậy một luật mờ dạng "If P then Q" sẽ được biểu diễn thành một quan hệ mờ R của phép kéo theo $P \Rightarrow Q$ với hàm thuộc của R trên không gian nền $U \times V$ được cho bởi phép kéo theo mà bạn dự định sử dụng:

$$R_{(A,B)}(u,v) = R_{P \Rightarrow Q}(u,v) = I(A(u),B(v)), \text{ với mọi } (u,v) \in U \times V.$$

Bây giờ quy trình suy diễn mờ đã có thể xác định:

Luật mờ (tri thức):	$P \Rightarrow Q$, với quan hệ cho bởi $I(A(u),B(v))$.
Sự kiện mờ (đầu vào):	$P' = \{x=A'\}$, xác định bởi tập mờ A' trên U
Kết luận:	$Q' = \{y=B'\}$

Sau khi đã chọn phép kéo theo I xác định quan hệ mờ $R_{(A,B)}$. B' là một tập mờ trên V với hàm thuộc của B' được tính bằng phép hợp thành $B' = A' \circ R_{(A,B)}$, cho bởi công thức:

$$B'(v) = \max_{u \in U}\{ \min(A'(u), I(A(u), B(v)))\}, \text{ với mỗi } v \in V.$$

1.4.3 Tiếp tục cách biểu diễn và diễn đạt như vậy, ta có thể xét dạng

"If P then Q else Q_1"

quen biết trong logic cổ điển và thường hay sử dụng trong các ngôn ngữ lập trình của ngành Tin học.

Có thể chọn những cách khác nhau diễn đạt mệnh đề này, sau đấy tìm hàm thuộc của biểu thức tương ứng. Chẳng hạn, chúng ta chọn

"If P then Q else Q_1" = $(P \wedge Q) \vee (\rceil P \wedge Q_1)$.

Thông thường Q và Q_1 là những mệnh đề trong cùng một không gian nền.

Giả thiết Q và Q_1 được biểu diễn bằng các tập mờ B và B_1 trên cùng không gian nền V, với các hàm thuộc tương ứng $B : V \rightarrow [0,1]$ và B_1: $V \rightarrow [0,1]$. Nếu Q và Q_1 không cùng không gian nền thì cũng sẽ xử lý tương tự nhưng với công thức phức tạp hơn.

Kí hiệu $R(P, Q, Q') = R(A, B, B_1)$ là quan hệ mờ trên $U \times V$ với hàm thuộc cho bởi biểu thức

$R(u,v) = \max\{\min(A(u), B(v)), \min(1-A(u), B_1(v))\}$, với mọi $(u,v) \in U \times V$.

Tiếp tục quy trình này chúng ta có thể xét những quy tắc lấy quyết định phức tạp hơn. Chẳng hạn chúng ta xét một quy tắc trong hệ thống mờ có 2 biến đầu vào và một đầu ra dạng

If A_1 and B_1 then C_1

else If A_2 and B_2 then C_2

else ...

1.4.4 Một dạng suy rộng khác trong cơ sở tri thức của nhiều hệ mờ thực tiễn, ví dụ điển hình là trong các hệ điều khiển mờ, có thể phát biểu dưới dạng sau:

Cho $x_1, x_2, \ldots, x_m$ là các biến vào của hệ thống, y là biến ra. Các tập A_{ij}, B_j, với $i = 1, \ldots, m$, $j = 1, \ldots, n$ là các tập mờ trong các không gian nền tương ứng của các biến vào và biến ra đang sử dụng của hệ thống, các R_j là các suy diễn mờ (các luật mờ) dạng "Nếu ... thì ..." (dạng if ... then)

R_1: Nếu x_1 là $A_{1,1}$ và ... và x_m là $A_{m,1}$ thì y là B_1

R_2: Nếu x_1 là $A_{1,2}$ và ... và x_m là $A_{m,2}$ thì y là B_2

...

R_n: Nếu x_1 là $A_{1,n}$ và ... và x_m là $A_{m,n}$ thì y là B_n

Bài toán

Cho:	Nếu x_1 là e_1* và ... và x_m là e_m*
Tính:	Giá trị y là u*

ở đây e_1*, ... , e_m* là các giá trị đầu vào hay sự kiện (có thể mờ hoặc giá trị rõ).

Chúng ta có thể nhận thấy rằng *phần cốt lõi của nhiều hệ mờ cho bởi cơ sở tri thức dạng R={các luật R_i} và các cơ chế suy diễn cài đặt trong mô tơ suy diễn.*

Tính toán quan hệ mờ cho những bộ luật phức tạp như thế các bạn có thể xem thêm công trình của M. Mizumoto và H.J. Zimmermann [8]. Những kiến thức về suy diễn mờ liên quan tới lập luận ngôn ngữ có thể đọc thêm bài của Nguyễn Cát Hồ [7] hay tài liệu [13]. Những bạn đọc nào quan tâm tới nghiên cứu toán học hiện đại trực tiếp liên quan tới logic mờ hãy xem thêm sách [10].

2 Các ứng dụng đa dạng

2.1 Sự phát triển của công nghệ mờ

Trong quá trình phát triển của Lý thuyết tập mờ và công nghệ mờ tại Nhật bản phải nhắc tới dự án lớn LIFE (the Laboratory for International Fuzzy Engineering) 1989–1995 do G.S. T.Terano (Tokyo Institute of Technology) làm Giám đốc điều hành – theo sáng kiến và sự tài trợ chính của Bộ ngoại thương và công nghiệp Nhật bản. Phòng thí nghiệm LIFE được thiết kế bởi G.S. M. Sugeno. Chính Giáo sư cũng đã thuyết phục được nhiều công ty công nghiệp hàng đầu của Nhật bản cung cấp tài chính và nhân lực, trở thành thành viên tập thể của dự án và chính họ trực tiếp biến các sản phẩm của phòng thí nghiệm thành sản phẩm hàng hoá.

Và kết quả là, theo Datapro, nền công nghiệp sử dụng công nghệ mờ của Nhật bản, năm 1993 có tổng doanh thu khoảng 650 triệu USD, thì tới năm 1997 đã ước lượng cỡ 6,1 tỷ USD và hiện nay hàng năm nền công nghiệp Nhật bản chi 500 triệu USD cho nghiên cứu và phát triển lý thuyết mờ và công nghệ mờ. Theo Giáo sư T. Terano [6] quá trình phát triển của công nghệ mờ có thể chia thành 4 giai đoạn sau:

1) *Giai đoạn* 1: Lợi dụng tri thức ở mức thấp.

Thực chất: Những ứng dụng trong công nghiệp chủ yếu là biểu diễn tri thức định lượng của con người.

Ví dụ điển hình: Điều khiển mờ.

Trong giai đoạn ban đầu nay, chủ yếu là cố gắng làm cho máy tính hiểu một số từ định lượng của con người vẫn quen dùng (như 'cao, nóng, ấm, yếu', v.v.). Một lí do rất đơn giản để đi tới phát triển điều khiển mờ là câu hỏi sau: *"Tại sao các máy móc đơn giản trong gia đình ai cũng điều khiển được mà máy tính lại không điều khiển được ?"*.

Có thể hầu hết các hệ điều khiển mờ là ở mức này. Thực tế tại mức ban đầu này đã đưa vào sử dụng rất nhiều loại máy mới có sử dụng logic mờ. *Đó là sự kiện rất quan trọng trong quá trình phát triển của logic mờ*, nhưng đó vẫn là các hệ thuộc giai đoạn 1.

2) *Giai đoạn* 2: Sử dụng tri thức ở mức cao.

Thực chất: Dùng logic mờ để biểu diễn tri thức.

Ví dụ: – Các hệ chuyên gia mờ.

– Các ứng dụng ngoài công nghiệp: y học, nông nghiệp, quản lý, xã hội học, môi trường.

Trong giai đoạn này cố gắng trang bị cho máy tính những tri thức cơ bản và sâu sắc hơn, những tri thức định tính mà trước tới nay chưa thể biểu diễn bằng định lượng, ví dụ như trong các hệ chuyên gia mờ. mô hình hoá nhiều bài toán khó trong quản lý các nhà máy mà trước đây chưa làm được.

3) *Giai đoạn* 3: Liên lạc–giao tiếp.

Thực chất: Giao lưu giữa người và máy tính thông qua ngôn ngữ tự nhiên.

Ví dụ: – Các robot thông minh.

– Các hệ hỗ trợ quyết định dạng đối thoại.

4) *Giai đoạn* 4: Trí tuệ nhân tạo tích hợp.

Thực chất: Giao lưu và tích hợp giữa trí tuệ nhân tạo ,logic mờ, mạng nơron và con người.

Ví dụ: – Giao lưu con người và máy tính.

– Các máy dịch thuật.

– Các hệ hỗ trợ lao động sáng tạo.

Giáo sư Terano còn cho rằng sự phát triển của công nghệ mờ và các hệ mờ tại Nhật bản đã và sẽ đi qua 4 giai đoạn trên.

2.2 Điều khiển mờ (Fuzzy Control)

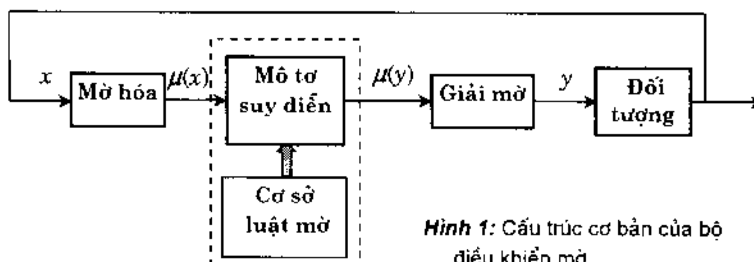
Như đã trình bày, những ứng dụng thực tiễn thành công nhất là Điều khiển mờ. Do vậy thật tự nhiên chúng ta sẽ trình bày khá chi tiết về lĩnh vực hấp dẫn này.

2.2.1 Cấu trúc cơ bản

Tư tưởng cơ bản của điều khiển dựa vào logic mờ là đưa các *kinh nghiệm chuyên gia* của những người vận hành giỏi hệ thống vào trong thiết kế các bộ điều khiển các quá trình trong đó quan hệ vào–ra (input-output) được cho bởi một tập các luật điều khiển mờ (dạng luật if…then).

Cấu trúc cơ bản (Basic architecture).

Cấu trúc cơ bản của một bộ điều khiển dựa vào logic mờ (fuzzy logic control FLC) gồm bốn thành phần chính (hình 1): *khâu mờ hoá* (a fuzzifier), một *cơ sở các luật mờ* (a fuzzy rule base), một *môtơ suydiễn* (an inference engine) và *khâu giải mờ* (a defuzzifier). Nếu đầu ra sau công đoạn giải mờ không phải là một tín hiệu điều khiển (thường gọi là tín hiệu điều chỉnh) thì chúng ta có một *hệ quyết định* trên cơ sở logic mờ.



Hình 1: Cấu trúc cơ bản của bộ điều khiển mờ.

2.2.2 Không gian Input-Output.

Vì mục tiêu của bộ điều khiển mờ là tính toán các giá trị của các biến điều khiển từ quan sát và đo lường các biến trạng thái của quá trình được điều khiển sao cho hệ thống vận hành như mong muốn. Như vậy việc chọn các biến trạng thái và các biến điều khiển phải đặc trưng cho các phép toán (the operator) của bộ điều khiển mờ và có tác động cơ bản lên sự quá trình thực hiện bộ FLC.

Kinh nghiệm của các chuyên gia và các tri thức về công nghệ đóng vai trò rất quan trọng trong việc lựa chọn các biến. Ví dụ các biến vào thường là trạng thái (state) sai lầm trạng thái (state error, state error derivate, state error integral . ...). Khi sử dụng biến ngôn ngữ. biến ngôn ngữ đầu vào x sẽ gồm các biến ngôn ngữ input x_i xác định trên không gian nền U_i và tương tự với biến đầu ra y gồm các biến ngôn ngữ output y_j trên không gian nền U_j. Khi đó

$$x = \{(x_i, U_i, \{A_{x_i}(1), \ldots, A_{x_i}(k_i)\}, \{\mu_{x_i}(1), \ldots, \mu_{x_i}(k_i)\} : i=1,2, \ldots, n\}$$

$$y = \{(y_i, V_i, \{A_{y_j}(1), \ldots, A_{y_j}(k_i)\}, \{\mu_{y_j}(1), \ldots, \mu_{y_j}(k_i)\} : i=1,2, \ldots, m\}$$

ở đây x_i là biến ngôn ngữ xác định trên không gian nền U_i, nhận từ – giá trị A_{x_i} với hàm thuộc $\mu_{x_i}(k)$ với $k = 1, 2, \ldots, k_i$. Tương tự cho các biến output y_j.

Ví dụ x_1 là biến tốc độ trên không gian nền là miền giá trị vật lý U_1 = [0, 200km/h]. Biến ngôn ngữ tốc độ có thể có các từ giá trị

{*rất chậm, chậm, trung bình, nhanh, rất nhanh*}.

Mỗi giá trị ngôn ngữ của biến này được xác định bằng một tập mờ trên U với các hàm thuộc $\mu_{chậm}(u), \ldots, \mu_{trung\ bình}(u)$.

2.2.3 Khâu mờ hoá.

Vì nhiều luật cho dưới dạng dùng các biến ngôn ngữ với các từ thông thường. Như vậy với những giá trị (rõ) quan sát được ,đo được cụ thể. để có thể tham gia vào quá trình điều khiển thì cần thiết phải mờ hoá.

Có thể định nghĩa. mờ hoá là một ánh xạ (mapping) từ không gian các giá trị quan sát được (rõ) vào không gian của các từ-tập mờ trên không gian nền của các biến ngôn ngữ input.

Ví dụ ứng với biến ngôn ngữ *tốc độ*, ta cho phép mờ hoá bằng ánh xạ

- Tốc độ một xe tải đo được: $u = 75$km/h.
- Từ đó có: $(\mu_{rất\ chậm}(75), \mu_{chậm}(75), \mu_{trung\ bình}(75), \mu_{nhanh}(75), \mu_{rất\ nhanh}(75))$.

2.2.4 Cơ sở các luật mờ

Dạng tổng quát của các luật điều khiển mờ là bộ các quy tắc mờ dạng IF .. . THEN. trong đó các điều kiện đầu vào và cả các biến ra (hệ quả) sử dụng các biến ngôn ngữ. Viết ở dạng tổng quát. cơ sở các luật mờ trong các hệ thống nhiều biến vào (output) và một biến ra (output) (tức là với các hệ MISO) cho dưới dạng sau:

Cho $x_1, x_2, \ldots, x_m$ là các biến vào của hệ thống. y là biến ra (thường là các biến ngôn ngữ). Các tập A_{ij}, B_j, với $i = 1, \ldots, m$, $j = 1, \ldots, n$ là các tập mờ trong các không gian nền tương ứng của các biến vào và biến ra đang sử dụng của hệ thống. Các R_j là các suy diễn mờ (các luật mờ) dạng "Nếu ... thì" (dạng if ...then)

R_1	Nếu x_1 là A_{11} và ... và x_m là A_{m1} thì y là B_1
R_2	Nếu x_1 là A_{12} và ... và x_m là A_{m2} thì y là B_2
⋮	⋮
R_n	Nếu x_1 là A_{1n} và ... và x_m là A_{mn} thì y là B_n
Cho:	Nếu x_1 là A_1* và ... và x_m là A_m*
Tính:	y là B*

ở đây $A_1^*, \ldots, A_m^*$ là các giá trị đầu vào hay sự kiện (có thể mờ hoặc giá trị rõ).

Một dạng tường minh các luật mờ thường cho dưới dạng

R_j : Nếu x_1 là A_{1j} và ... và x_m là A_{mj} thì $y=f_j(x_1, x_2, \ldots, x_m)$,

$j=1, \ldots, n$. Ở đây $f_j(x_1, x_2, \ldots, x_m)$ là một hàm của các biến trạng thái.

2.2.5 Mô tơ suy diễn

Đây là phần cốt lõi nhất của FLC trong quá trình mô hình hoá các bài toán điều khiển và chọn quyết định của con người trong khuôn khổ vận dụng logic mờ và lập luận xấp xỉ. Do các hệ thống được xét dưới dạng hệ vào–ra nên luật suy diễn *modus ponens suy rộng* đóng một vai trò rất quan trọng.

Như đã trình bày kỹ trong phần trước. Suy luận xấp xỉ, phép hợp thành và phép kéo theo của logic mờ sẽ quyết định những công việc chính trong quá trình tính toán cũng như trong quá trình rút ra kết luận. Bảng sau đây giới thiệu một số phép kéo theo mờ (*fuzzy implications*) thường được sử dụng trong diễn đạt các luật mờ.

Phép kéo theo	Công thức
Toán tử min [Mamdani]	$a \Rightarrow b = \min(a, b)$
Toán tử tích [Larsen]	$a \Rightarrow b = a.b$
Tích bị chặn	$a \Rightarrow b = \max(0, a+b-1)$
Quy tắc số học [Zadeh]	$a \Rightarrow b = \min(1, 1-a+b)$
Quy tắc max–min[Zadeh]	$a \Rightarrow b = \max(1-a, \min(a,b))$
Suy diễn bình thường	$a \Rightarrow b = \begin{cases} 1 & \text{nếu } a \le b \\ 0 & \text{nếu } a > b \end{cases}$
Logic Boole	$a \Rightarrow b = \max(1-a, b)$
Logic Godel	$a \Rightarrow b = \begin{cases} 1 & \text{nếu } a \le b \\ b & \text{nếu } a > b \end{cases}$
Phép kéo theo Goguen	$a \Rightarrow b = \begin{cases} 1 & \text{nếu } a \le b \\ b/a & \text{nếu } a > b \end{cases}$

2.2.6 Khâu giải mờ

Đây là khâu thực hiện quá trình xác định một gía trị rõ có thể chấp nhận được làm đầu ra từ hàm thuộc của giá trị mờ đầu ra. Có hai phương pháp giải mờ chính: *Phương pháp cực đại* và *phươngpháp điểm trọng tâm*. Tính toán theo các phương pháp này không phức tạp. Có thể tham khảo ở các tài liệu [3,12].

2.2.7 Ứng dụng

Ứng dụng đầu tiên của điều khiển mờ phải kể đến của nhóm Mamdani và Assilian năm 1974,[14]. Từ đấy phạm vi ứng dụng thực tiễn của điều khiển mờ trong các lĩnh vực khác nhau đã hết sức rộng: từ điều khiển lò nung xi măng [Larsen,1980- đây là ứng dụng thực sự đầu tiên vào sản xuất công nghiệp], quản lý các bãi đỗ xe [Sugeno và cộng sự 1984,1985.

1989], điều khiển vận hành hệ thống giao thông ngầm, quản lý nhóm các thang máy [Fujitec,1988], điều chỉnh việc hoà clo trong các nhà máy lọc nước, điều khiển hệ thống máy bơm làm sạch nước [Yagishita et al., 1985], điều khiển hệ thống năng lượng và điều khiển phản ứng hạt nhân [Bernard,1988, Kinoshita et al., 1988], máy bay trực thăng [Sugeno, 1990], v.v..., cho tới thám sát các sự cố trên đường cao tốc [Hsiao et al., 1993] các thiết bị phần cứng mờ [fuzzy hardware devices, Togai và Watanabe, 1986, nhóm cộng tác với GS. Yamakawa, 1986, 1987,1988 ...].

Trong số những ứng dụng thực sự thành công trong thực tiễn còn phải nhắc tới tới bộ FLC dùng trong quản lý sân bay [Clymer et al. ,1992], các hệ thống điều khiển đường sắt và các hệ thống cần cẩu container [Yasunobu và Miyamoto, 1985, Yasunobu et al., 1986, 1987]. Một ứng dụng rất hay của điều khiển mờ là hệ điều khiển "*the camera tracking control system*" của NASA ,1992 ....

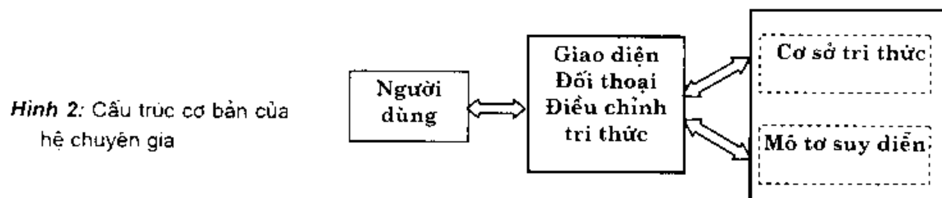
Chúng ta cũng không thể không nhắc tới các máy móc trong gia đình dùng FLC đang bán trên thị trường thế giới: máy điều hoà nhiệt độ [hãng Mitsubishi], máy giặt [Matsushita, Hitachi, Sanyo], các video camera [Sanyo, Matsushita], tivi, camera [hãng Canon], máy hút bụi, lò sấy (microwave oven) [Toshiba] vv....

Ngay từ 1990, trong một bài đăng ở tạp chí AI Expert, Vol.5, T.J. Schwartz đã viết: "Tại Nhật bản đã có hơn 120 ứng dụng của điều khiển mờ ".

2.3 Các hệ chuyên gia mờ (Fuzzy Expert Systems)

2.3.1 Các hệ chuyên gia (Expert Systems) là một nhánh của bộ môn Trí tuệ nhân tạo (Artifical Intelligence) sử dụng các trí thức chuyên biệt để giải quyết bài toán ở giai đoạn dùng các chuyên gia – con người. Chúng phát triển vào những năm 1970 và đã được ứng dụng trong khá nhiều lĩnh vực. Ngày nay nói đến hệ chuyên gia thì thực chất hiểu là các hệ thống trong đó có sử dụng *công nghệ hệ chuyên gia* (expert system technology) bao gồm: các ngôn ngữ hệ chuyên gia chuyên dụng, các chương trình và cả các phần cứng được thiết kế nhằm phát triển và vận hành các hệ chuyên gia.

Hiện nay trong các sách báo người ta thường dùng từ đồng nghĩa là "*hệ chuyên gia trên cơ sở tri thức* " (knowledge–based expert system). Sau đây chúng ta sẽ cho cấu trúc cơ bản (hình 2) và cách sử dụng hệ chuyên gia.



Hình 2: Cấu trúc cơ bản của hệ chuyên gia

Dưới đây là vài cột mốc chính trong quá trình hình thành và phát triển của các hệ chuyên gia:

- 1957 Bắt đầu "Chương trình giải quyết bài toán tổng quát" (GPS).
- 1958 Ngôn ngữ lập trình LISP.
- 1965 DENDRAL- hệ chuyên gia đầu tiên.
- 1968 Mạng ngữ nghĩa.
- 1970 Ngôn ngữ lập trình PROLOG.
- 1973 MYCIN - hệ chuyên gia dành cho chuẩn đoán y học (Shortliffe et all.).
- 1976 Hệ chuyên gia PROSPECTOR trong lĩnh vực địa chất.
- 1985 Công cụ hệ chuyên gia CLIPS của NASA (expert system tool).

Trong số những hệ ban đầu này hệ MYCIN có ý nghĩa rất lớn vì mấy lí do sau :

a) Nó chỉ ra rằng Trí tuệ nhân tạo có thể sử dụng trong các bài toán thực tiễn.

b) Trong MYCIN đã đưa vào nhiều khái niệm mới rất có lợi cho những phát triển về sau.

c) Nó chứng minh tính khả thi của các khung của hệ chuyên gia, các công cụ tạo lập hệ chuyên gia (expert system *shell*).

2.3.2 Bên cạnh những thành công của công nghệ hệ chuyên gia. lý thuyết tập mờ và logic mờ có nhiều ưu điểm trong biểu diễn tri thức của các chuyên gia. Cho nên việc đưa các luật mờ và đặc biệt các biến ngôn ngữ và hàm thuộc đã xuất hiện khá sớm. Các hệ chuyên gia trình bày dưới đây đã sử dụng các luật mờ (fuzzy rules).

Tên/ tác giả/ năm	Lĩnh vực
CADIAG-2,Adlassnig et al.,1985	Y (internal medicine)
EMERGE. Hudson, Cohen,1988	Phân tích đau ngực
ESP. Zimmermann. 1989	Kế hoạch mức chiến lược
FAULT, Whalen et al., 1987	Kế toán
OPAL. Bensana et al., 1988	Lập lịch công việc

Thực tiễn đã dẫn tới cần phối hợp tốt hơn hai loại công nghệ này. đó là nhu cầu về nghiên cứu các *hệ chuyên gia mờ* (fuzzy expert systems). Những nghiên cứu sau đây là ví dụ:

- FESS - một hệ chuyên gia mờ tái sử dụng ,Hall và Kandell, 1992
- Hệ chuyên gia mờ có mục đích tổng quát, Schneider và Kandel, 1994.
- Những khung cho hệ chuyên gia mờ. Umano, Hatono và Tamura (Fuzzy expert system shells), 1994.
- Công trình của Whalen và Schott. 1992. tạo ra mạng suy diễn ngôn ngữ mờ (Fuzzy linguistic inference network generator).

Chúng ta cũng không khó khăn chỉ ra các ứng dụng thực tiễn của các hệ chuyên gia mờ. Sau đây là vài ví dụ:

- Von Altrock và Krause đã phát triển hệ chuyên gia mờ dành cho công nghiệp ôtô của hãng Mercedes-Benz tại CHLB Đức, 1994 (xem bài báo Multi-criteria decision making in the German automotive industry).
- Hệ EMERGE của Hudsson và Cohen, 1992 là hệ chuyên gia y học trên cơ sở các luật mờ trợ giúp cho các phòng cấp cứu và đưa các lời chỉ dẫn về kiểm tra các hiện tượng đau.
- Hệ EAR của López de Mántaras et al. dành cho chẩn đoán y học, còn hệ FLING, của Whalen và Schott, 1992 thì trợ giúp cho công việc nắm bắt , phân tích tri thức.
- Về các hệ MedFrame/CADIAG-IV, 1996, FuzzyKBWean, 1998, và Fuzzy-ARDS, 1999, đang sử dụng tại Viên, C.H. Áo có thể tìm thấy trong bài báo của K.P. Adlassnig [23, trang 134-138].

Những tri thức cơ bản và đủ hiện đại về hệ chuyên gia có thể tìm thấy trong sách [18].

2.4 Nhận dạng mờ (Fuzzy Pattern Recognition)

2.4.1 Bài toán nhận dạng.

Nhiều thông tin hàng ngày chúng ta cần thu nhận và xử lý nhiều khi có hình dạng phức tạp. Bộ môn nhận dạng cung cấp cho chúng ta một số phương pháp để phát hiện cấu trúc chính và một số nét cơ bản của các hình dạng đó. Các tiếp cận chính tới các bài toán nhận dạng là: *phân lớp tuyến tính, tiếp cận thống kê, lý thuyết mờ, các perceptrons, phân lớp cơ sở tri thức dựa vào kĩ thuật của trí tuệ nhân tạo.*

Nghĩ tới tiếp cận từ Lý thuyết mờ là hoàn toàn tự nhiên vì các hình dạng quan sát được, các thông tin nhận được thường mờ, thậm chí có thể nói là "rất mờ". Một số kết quả đã được công bố (Ví dụ: Bezdek, 1981, Pedrrycz, 1990, hay Bezdek và Pal, 1992) và đã ứng dụng thành công trong xử lý ảnh, nhận dạng tiếng nói,trong robot thông minh, nhận dạng các đối tượng hình học .... Hình 3 trình bày sơ đồ chung của một quá trình nhận dạng.



Hình 3: Quá trình nhận dạng

Bây giờ hãy nói thêm về một số khái niệm.

Dữ liệu (data) thu được từ các quá trình vật lý hoặc các hiện tượng. Dữ liệu (hiểu theo nghĩa rộng) có thể ở dạng định tính, định lượng, dạng số, hình, hay một đoạn văn, ngôn ngữ hay tổ hợp của các dạng trên.

Không gian các dáng (pattern space) - còn được gọi là cấu trúc (structure), đó là tập các dáng vẻ, các kiểu dáng trong đó thông tin được tổ chức sao cho chúng có thể góp phần phát hiện mối liên hệ giữa các biến. Nói chung số chiều của không gian các dáng cần ít hơn số chiều của không gian dữ liệu.

Không gian các đặc tả (feature space) là không gian nằm giữa không gian dữ liệu và không gian phân lớp. Nói chung số chiều của không gian đặc tả thường phải ít hơn nhiều số chiều của không gian dữ liệu. Vấn đề khó khăn là chọn những đặc tả nào cho đủ phản ánh những nét quan trọng nhất của đối tượng.

Không gian phân lớp dĩ nhiên phải chứa được các lớp mà người nghiên cứu muốn phân. Tuy nhiên thực chất có bao nhiêu lớp phân biệt thì nhiều khi ngay từ đầu cũng khó xác định.

Một nhận xét cũng nên để ý là tính mờ không chỉ trong không gian dữ liệu mà còn nằm trong không gian các đặc tả và cả trong các thuật toán phân lớp.

2.4.2 Phân nhóm và vai trò trong thực tế

Theo [17] các chuyên gia hiện đang nghiên cứu về nhận dạng mờ có thể chia theo 4 nhóm đề tài chính:

- Nhận dạng tĩnh các đối tượng 3 chiều, có màu.
- Nhận dạng động các đối tượng 3 chiều, có màu.
- Nhận dạng theo công nghệ hiển thị (visual recognition) các đối tượng 3 chiều, có màu.
- Nhận dạng động các biểu hiện của con người (human expressions).

Ai cũng hiểu được tầm quan trọng của các đề tài nhận dạng các đối tượng động 3-chiều trong quốc phòng, an ninh và ngành viễn thám. Nhưng ngay đối với các ngành dân sự độ quan trọng của các đề tài này cũng được phản ảnh qua bảng tổng kết sau cuộc thăm dò ý kiến của hơn 200 chuyên gia hàng đầu về công nghệ mờ tiến hành 1989 bởi hai giáo sư A. Ishikawwa và T. L. Wilson [17] (các con số ghi trong bảng tính theo % của các ý kiến).

Bảng 1

Đề tài	Độ quan trọng		
	Cao	Trung bình	Thấp
1) Nhận dạng tĩnh dùng công nghệ mờ			
– Các hệ robot sản xuất dùng công nghệ hiển thị	84.0	16.0	0
– Các hệ theo dõi tự động dùng công nghệ hiển thị	80.8	19.2	0
– Các hệ vận tải robot dùng công nghệ hiển thị	42.9	47.6	9.5
2) Nhận dạng động dùng công nghệ mờ			
– Các hệ robot sản xuất dùng công nghệ hiển thị	81.8	9.1	9.1
– Các hệ theo dõi tự động dùng công nghệ hiển thị	78.3	13.0	8.7

2.5 Hệ hỗ trợ quyết định và bài toán lấy quyết định

Trong phần áp chót này chúng ta sẽ dành cho hai lớp bài toán sau: bắt đầu với một hệ hỗ trợ quyết định mờ và sau đó là một mô hình của bài toán lấy quyết định tập thể.

2.5.1 Các hệ hỗ trợ quyết định (Decision Support Systems) đã đi qua một chặng đường khá dài, từ những hệ thông tin quản lý và những cơ sở dữ liệu được tổ chức khoa học và hoàn chỉnh hơn nhằm phục vụ cho nhiều nhiệm vụ quan trọng và thường là tương đối khó trong các doanh nghiệp, tới giai đoạn xây thêm các hệ chuyên gia ,coi đó là một phân hệ quan trọng giúp cho người lấy quyết định thêm những lời khuyên quý gía.

Đưa các thành tựu của logic mờ và nhiều tri thức mới của lý thuyết tập mờ vào hệ hỗ trợ quyết định là một tiếp cận tự nhiên. Trong giai đoạn sau này các thành tựu hiện đại này thường được đưa vào trong các hệ hỗ trợ quyết định nhằm góp phần tìm lời giải đáp cho nhiều lớp bài toán phức tạp trong các tình huống có nhiều yếu tố bất định , nhiều tri thức ít xác định, hoặc chưa được phát biểu chính xác.

2.5.2 Để đơn gián chúng ta hãy xem bài báo giới thiệu về hệ hỗ trợ quyết định FOREX của hai tác giả Satoru Fukami (NTT Data Communication Systems Cor.) và Minoru Yoneda (Mitsubishi Kasei Cor.)[16 ,trang 17-66].

FOREX là tên viết tắt của FOReign EXchange dealing support system - một trong 9 đề tài chính trong giai đoạn 1 (3 năm đầu) của LIFE nhằm hỗ trợ việc buôn bán trao đổi ngoại tệ bằng những dự báo các tỷ giá hối đoái, dựa trên các ước lượng bằng kinh nghiệm của các chuyên gia nhằm nâng cao hiệu quả của cả hệ thống kinh doanh.

Để cho hệ hoạt động có hiệu quả cần thu thập các thông tin số (dạng numeric data), các dữ liệu dạng văn bảng(text data), những gì liên quan tới các giao dịch ngoại hối, cũng như các kinh nghiệm của các chuyên gia. Các bài toán dự báo được giải quyết chủ yếu dựa vào lý thuyết tập mờ.

Sau đây là vài thông tin về cấu hình của hệ thống:

- Trước hết tại LIFE đã phát triển LIFE FEShell ("LIFE Fuzzy Expert System Shell"). Hệ thống này đã được sữ dụng như một môi trường hỗ trợ và là công cụ tạo lập hệ chuyên gia.
- FOREX được viết bởi Commom LISP và ngôn ngữ C.
- Phần cứng đã sử dụng là một SUN-4 workstation.

FOREX công bố bởi LIFE năm 1991 đã là một hệ khá lớn, nó chứa tới 5100 luật.

Tài liệu trích dẫn

[1] **L.A. Zadeh**: Fuzzy sets, Inform. and Control .8, 1965. 338-353.

[2] **D. Dubois and H. Prade**: Fuzzy Sets and Systems. Academic Press, N.Y. , 1980.

[3] **H.J. Zimmermann**: Fuzzy Set Theory - and Its Applications.2 nd Ed., Kluwer Acad. Pub., Dodrech. 1991.

[4] **L.A.Zadeh:** The concept of a linguistic variable and its application to approximate reasoning, American Elsevier Pub. company, New York. 1973.

[5] **L.A. Zadeh:** Fuzzy sets as a basis for a theory of possibility, Fuzzy Sets and Systems, 1. 1978, 3-28.

[6] **E.P. Klement:** Some mathematical aspects of fuzzy sets : Triangular norms, fuzzy logics, and generalized measures. Fuzzy sets and systems. 90, 1997, 133-140.

[7] **N. Cat Ho:** A method in linguistic reasoning on a knowledge base representing by sentences with linguistic belief degree,Fund. Informaticae. 28, 1996, 247- 260.

[8] **M.Mizumoto, H.J.Zimmermann:** Comparison of fuzzy reasoning methods, Fuzzy Sets and Systems, 8 ,1982, 253-283.

[9] Fuzzy reasoning in information, decision and control systems. Ed. S.G. Tzafestas and A. N.Venetnopoulos. Kluwer Acad. Pub., Dordrecht.1994.

[10] Non-clssical logics and their applications to fuzzy subsets, Ed. U. Hohle and E.P. Klement, Kluwer Acad. Pub., Dordrecht.1995.

[11] **R.R. Yager:** Fuzzy logics and artificial intelligence .Fuzzy Sets and Systems, 90, 1997. 193-198 .

[12] **Phan Xuân Minh và Nguyễn Doãn Phước:** Lý thuyết điều khiển mờ.In lần 2. NXB Khoa học và Kỹ thuật, 1999, Hà nội.

[13] Hệ mờ và ứng dụng. Biên tập tập thể : Nguyễn Hoàng Phương, Bùi Công Cường, Nguyễn Doãn Phước, Phan Xuân Minh và Chu Văn Hỷ. NXB Khoa học và Kỹ thuật, 1998, Hà nội.

[14] **E.H.Mamdani:** Application of fuzzy algorithms for the control of a simple dynamic plant, Proc. IEEE .121,(12).p. 1585-1588,1974.

[15] **Kazuyuki Suzuki:** Fuzzy modeling and process control system design, in [14], p.103-136.

[16] **Anca L. Ralescu, Ed.:** Applied research in fuzzy technology, Kluwer Acad. Pub., Dordrecht, 1994.

[17] **A. Ishikawa and T.L. Wilson:** Analysis and Evaluation of Fuzzy Systems, Kluwer Acad. Pub., Dordrecht, 1995.

[18] **Giarratano and G.Riley:** Expert Systems. Principles and Programming, Second Edition, PWS Pub. Com., Boston. 1994.

[19] **Satoru Fukami and Minoru Yoneda:** Decision Support System, [16] trang 17-66.

[20] **C.T.Lin and C.S.George Lee:** Neural Fuzzy Systems, Prentice-Hall Inter. Inc.,New Jersey, 1996.

[21] **C.H.Chen (Ed.):** Fuzzy Logic and Neural Network Handbook, McGraw-Hill,Inc., New York,1996.

[22] **T.Yamakawa and G. Matsumoto (Eds.):** Methodologies for the Conception, Design and Application of Soft Computing, *Proceedings of the 5th International Conference on Soft Computing and Information/Intelligent Systems*, IIZUKA 98. Vol. 1 and Vol 2, World Scientific, New Jersey.1998.

[23] *Proceedings of the International Symposium on Medical Informatics and Fuzzy Technology*, MIF 99. August 26-29,1999, Trung tâm KHTN và CNQG, Hà nội,1999.

[24] **Bùi công Cường:** Cơ sở toán học của các hệ mờ,Giáo trình, ĐH Bách khoa Hà nội, 1998-1999.

[25] **Bùi công Cường:** Một số kiến thức cơ sở của logic mờ trong các hệ mờ,trong [13], trang 1-21.

[26] **Bùi công Cường:** A multiple criteria group decision making model under linguistic assessments,*Proceedings of the International Symposium on Medical Informatics and Fuzzy Technology*,MIF 99. August 26-29,1999, Trung tâm KHTN và CNQG, Hà nội ,1999, 291-297.

4 NHẬP MÔN ĐIỀU KHIỂN MỜ

Nguyễn Doãn Phước & Phan Xuân Minh
Đại học Bách khoa Hà Nội

Có lẽ hầu hết mọi người hiện nay không ai là chưa từng nghe đến khái niệm điều khiển mờ (*Fuzzy control*) cũng như tên các thiết bị điều khiển được tích hợp dựa trên nguyên lý tập mờ (*Fuzzy set*). Những thiết bị làm việc trên cơ sở lý thuyết tập mờ hiện có khắp mọi nơi trong cuộc sống thường nhật như máy giặt Fuzzy, máy ảnh Fuzzy, bàn là Fuzzy, nồi cơm điện Fuzzy ... đã giúp cho sự phổ thông hóa đó của những khái niệm lý thuyết này.

Nhìn lại quãng đường đã đi, kể từ thời điểm ra đời của lý thuyết tập mờ vào khoảng giữa thập niên 60 do nhà toán học người Mỹ *Zahde* đưa ra nhằm thay thế, đơn giản hóa các khái niệm đầy tính lý thuyết của xác suất, của quá trình ngẫu nhiên, thì cho tới ngày nay, điều khiển mờ đã có những bước phát triển vượt bậc, đóng góp không nhỏ vào sự tăng trưởng, hiện đại hóa cuộc sống con người. Những khái niệm của điều khiển mờ mà trước đây còn mang đầy tính trừu tượng thì nay nó đã được đưa vào ngôn ngữ cộng đồng như một sự đương nhiên ai cũng biết hoặc cũng được nghe đến một cách thường xuyên nhờ các phương tiện của thông tin đại chúng như báo, đài, truyền hình quảng cáo .... Sự phát triển nhanh mang tính vượt bậc của *điều khiển mờ* có nguyên nhân của nó:

- Thứ nhất là trên cơ sở suy luận mờ, nguyên lý điều khiển mờ đã cho phép con người tự động hóa được kinh nghiệm điều khiển của họ cho một quá trình, một thiết bị ... . tạo ra được những bộ điều khiển làm việc tin cậy thay thế được họ song vẫn mang lại chất lượng như họ đã từng đạt được.
- Thứ hai là với nguyên tắc mờ, bộ điều khiển tổng hợp được có cấu trúc đơn giản đến kỳ lạ so với những bộ điều khiển kinh điển khác có cùng chức năng. Sự đơn giản đó đã đóng vai trò quan trọng trong việc tăng độ tin cậy cho thiết bị, giảm giá thành sản phẩm.
- Và cuối cùng nhưng không kém phần quan trọng cho sự phát triển vượt bậc đó của điều khiển mờ là những cải tiến liên tiếp của kỹ thuật vi xử lý, một cầu nối không thể thiếu giữa kết quả nguyên cứu của lý thuyết điều khiển mờ với thực tế ứng dụng.

Trong bài này chúng tôi mong muốn cùng bạn đọc nhìn lại một cách tổng quan có tính hệ thống về nguyên lý làm việc của một bộ điều khiển mờ để có thể tự tạo ra được, tự tổng hợp được các thiết bị tự động điều khiển trên nguyên lý tập mờ chứ không đơn thuần là chỉ sử dụng chúng.

Để thực hiện được mục đích đặt ra đó, ta sẽ lần lượt đi qua các phần sau:

- Trước hết ta sẽ nghiên cứu xem bộ điều khiển mờ làm việc theo nguyên lý cơ bản nào khác so với những "*bộ điều khiển không mờ*". Trong phần này ta sẽ làm quen với các

khái niệm được dùng đến ở những phần sau là *biến ngôn ngữ, giá trị ngôn ngữ, luật hợp thành* và *mệnh đề hợp thành*.

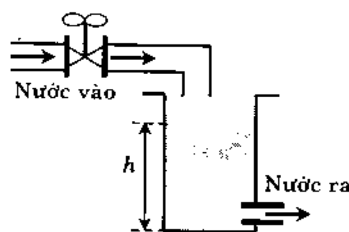
- Tiếp theo là phần giới thiệu *lý thuyết tập mờ* dưới góc nhìn của một người làm điều khiển. Tại đây chúng ta sẽ tiến hành mô tả chi tiết khái niệm giá trị ngôn ngữ, *phép suy diễn mờ* (còn gọi là *phép kéo theo*) để có thể cài đặt luật hợp thành trong bộ điều khiển mờ.
- Phần thứ ba, chúng ta sẽ làm quen với *thiết bị hợp thành* có nhiệm vụ thực hiện luật hợp thành được xem như là một "*phương châm hành động*" của bộ điều khiển mờ.
- Cuối cùng chúng ta sẽ cùng nhau xây dựng một bộ điều khiển mờ hoàn chỉnh với những công đoạn bổ sung thêm bao gồm *mờ hóa* và *giải mờ*.

1 Nguyên lý làm việc

Trong rất nhiều các bài toán điều khiển, khi mà đối tượng không thể mô tả bởi một mô hình toán học hoặc có thể mô tả được song mô hình của nó lại quá phức tạp, cồng kềnh, không ứng dụng được, thì điều khiển mờ chiếm ưu thế rõ rệt. Ngay cả ở những bài toán đã điều khiển thành công theo nguyên tắc kinh điển thì việc áp dụng điều khiển mờ cũng sẽ vẫn mang lại cho hệ thống sự cải tiến về tính đơn giản, gọn nhẹ.

Lý do chính dẫn tới suy nghĩ áp dụng logic mờ để điều khiển nằm ở chỗ trong rất nhiều trường hợp, con người chỉ cần dựa vào kinh nghiệm (hoặc ý kiến chuyên gia) vẫn có thể điều khiển được đối tượng cho dù đối tượng có *thông số kỹ thuật không đúng hoặc thường xuyên bị thay đổi ngẫu nhiên và do đó mô hình toán học của đối tượng điều khiển không chính xác*, đó là chưa nói tới chúng có thể hoàn toàn sai. Việc điều khiển theo kinh nghiệm như vậy, có thể bị đánh giá là không chính xác như các yêu cầu kỹ thuật đề ra (ví dụ như điều khiển tối ưu), song đã giải quyết được vấn đề trước mắt là vẫn đảm bảo được về mặt định tính các chỉ tiêu chất lượng định trước.

Ta hãy xét một ví dụ đơn giản là điều khiển mực nước. Hình 1 mô tả nguyên lý của bài toán. Không phụ thuộc vào lượng nước chảy ra khỏi bình ta phải chỉnh van cho lượng nước chảy vào bình vừa đủ để sao cho mực nước trong bình là h luôn không đổi. Tất nhiên rằng bài toán điều khiển này đã được giải quyết rất đơn giản và ta có thể bắt gặp nó trong những thiết bị gia đình thông dụng. Nhưng ở đây ta đề cập lại nó từ phương diện điều khiển mờ để thông qua đó hiểu rõ hơn về bản chất của một bộ điều khiển mờ.



Hình 1: Điều khiển mực nước.

Hình dung bộ điều khiển là con người. Vậy thì con người sẽ điều chỉnh van đóng/mở nước vào như thế nào?. Ta có thể dựa vào kinh nghiệm để nói rằng họ sẽ điều chỉnh van theo bốn nguyên tắc như sau:

a) Nếu *mực nước* là *thấp nhiều* thì *van* ở mức độ mở *to*.

b) Nếu *mực nước* là *thấp ít* thì *van* ở mức độ mở *nhỏ*.

c) *Nếu mực nước* là *cao* thì *van* ở vị trí *đóng*.

d) Nếu *mực nước* là *đủ* thì *van* ở vị trí *đóng*.

Một bộ điều khiển làm việc theo luật như trên để thay thế con người sẽ được gọi là *bộ điều khiển mờ*. Khác hẳn với những phương pháp kinh điển, điều khiển mờ không cần đến mô hình toán học của đối tượng.

Bốn nguyên tắc điều khiển trên, trong điều khiển mờ được gọi là bốn *mệnh đề hợp thành*. Kinh nghiệm điều khiển mực nước chung gồm cả bốn nguyên tắc đó được gọi là *luật hợp thành*.

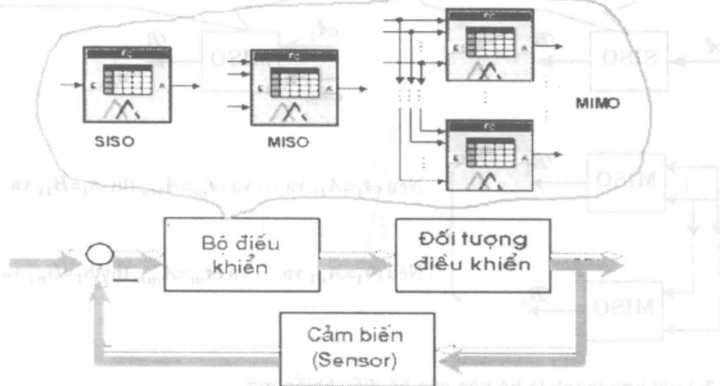
Bên cạnh hai khái niệm luật hợp thành và mệnh đề hợp thành vừa được trình bày, trong ví dụ về bốn nguyên tắc điều khiển ta còn thấy những tên gọi khác như *mực nước* và *van*. Chúng chính là các tín hiệu vào (mực nước) và ra (van) của bộ điều khiển (mà ở đây là con người). Những tín hiệu vào và ra này được gọi chung lại thành *biến ngôn ngữ*.

Mỗi biến ngôn ngữ lại có nhiều giá trị. Chẳng hạn trong ví dụ trên thì:

– biến ngôn ngữ *mực nước* có bốn giá trị là *thấp nhiều*, *thấp ít*, *đủ*, *cao*.

– hoặc biến *van* có ba giá trị là *to*, *nhỏ*, *đóng*.

Những giá trị này của các biến ngôn ngữ được gọi chung lại là *giá trị ngôn ngữ*.



Hình 2: Phân nhóm các bộ điều khiển theo số tín hiệu vào/ra.

Như vậy, bộ điều khiển mờ có thể được hiểu là một bộ điều khiển làm việc theo nguyên tắc *tự động hoá những kinh nghiệm điều khiển của con người*. Những kinh nghiệm này phải được đúc kết lại *luật hợp thành* gồm nhiều *mệnh đề hợp thành* với cấu trúc chung như sau:

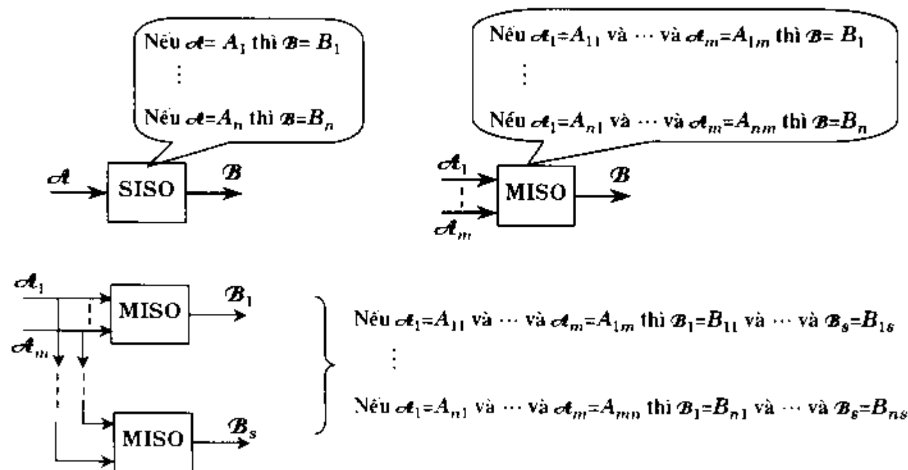
Nếu $\mathcal{A}=A_i$ **thì** $\mathcal{B}=B_j$. (1)

trong đó $\mathcal{A}$ là biến ngôn ngữ đầu vào, $\mathcal{B}$ là biến ngôn ngữ đầu ra, A_i , i=1,2, … là các giá trị ngôn ngữ của biến $\mathcal{A}$ và B_j , j=1,2, … là các giá trị ngôn ngữ của biến $\mathcal{B}$.

Một bộ điều khiển mờ chỉ có một tín hiệu vào ra như ta đã xét được gọi là bộ điều khiển SISO (*Single Input, Single Output*). Song tất nhiên một bộ điều khiển mờ không nhất thiết là chỉ có một tín hiệu vào và một tín hiệu ra. Nó có thể có rất nhiều tín hiệu đầu vào cũng như có nhiều tín hiệu ra. Những bộ điều khiển mờ có nhiều đầu vào/ra như vậy được gọi là bộ điều khiển MIMO (*Multi Input, Multi Output*). Nói cách khác cũng giống như một bộ điều khiển kinh điển, một bộ điều khiển mờ cũng có thể có nhiều tín hiệu vào và nhiều tín hiệu ra. Ta phân chia chúng thành các nhóm:

- Nhóm bộ điều khiển SISO nếu nó chỉ có một đầu vào và một đầu ra.
- Nhóm MIMO nếu chúng có nhiều đầu vào và nhiều đầu ra.
- Nhóm bộ điều khiển SIMO nếu nó chỉ có một đầu vào nhưng nhiều đầu ra.
- Nhóm MISO nếu chúng có một đầu vào và nhiều đầu ra.

Hình 2 mô tả trực quan các nhóm bộ điều khiển mờ này.



Hình 3: Luật hợp thành là bộ não của bộ điều khiển mờ.

Nếu một bộ điều khiển mờ có nhiều tín hiệu vào/ra thì tương ứng mệnh đề hợp thành của nó cũng phải có nhiều biến ngôn ngữ vào $\mathcal{A}_1$, $\mathcal{A}_2$, … $\mathcal{A}_m$ và nhiều biến ngôn ngữ ra $\mathcal{B}_1$, $\mathcal{B}_2$, … $\mathcal{B}_s$. Từng biến ngôn ngữ đó lại có nhiều giá trị ngôn ngữ. Ta ký hiệu A_{ki} , i=1,2, … là một giá trị của biến $\mathcal{A}_k$, k=1,2, … , m cũng như B_{jl} , l=1,2, … là một giá trị của biến $\mathcal{B}_j$, j=1,2, … , s thì mệnh đề hợp thành của nó sẽ có dạng

$$\textbf{Nếu } \mathcal{A}_1=A_{i1} \text{ và } \cdots \text{ và } \mathcal{A}_m=A_{im} \textbf{ thì } \mathcal{B}_1=B_{i1} \text{ và } \cdots \text{ và } \mathcal{B}_s=B_{is}\,. \qquad (2)$$

Bộ não của điều khiển mờ là luật hợp thành. Luật hợp thành của bộ điều khiển mờ SISO với các mệnh đề hợp thành dạng (1) được gọi là *luật hợp thành đơn*. Ngược lại luật hợp thành có các mệnh đề dạng (2) của bộ điều khiển mờ MIMO có tên là *luật hợp thành kép*.

Ta sẽ dành riêng cho luật hợp thành MISO có mệnh đề theo cấu trúc:

$$\textbf{Nếu } \mathcal{A}_1=A_{i1} \text{ và } \cdots \text{ và } \mathcal{A}_m=A_{im} \textbf{ thì } \mathcal{B}=B_l\,. \qquad (3)$$

của bộ điều khiển có nhiều tín hiệu vào nhưng chỉ có một tín hiệu ra, một sự quan tâm đặc biệt. Lý do nằm ở chỗ mọi luật hợp thành kép (2) đều có thể được đưa về dạng hợp song song của nhiều luật hợp thành MISO (hình 3).

2 Lý thuyết tập mờ trong điều khiển

2.1 Định nghĩa tập mờ

Quay lại ví dụ về điều khiển mực nước đã nói tới ở phần 1 với những giá trị ngôn ngữ *thấp nhiều*, *thấp ít*, *đủ*, *cao* của biến đầu vào mực nước, cũng như *to*, *nhỏ*, *đóng* của biến đầu ra là van. Các giá trị đó sẽ gây ra nhiều cảm giác phân vân cho người thiết kế bộ điều khiển nếu không đưa vào đó khái niệm *tập mờ*.

Tại sao lại như vậy?. Để trả lời ta giả sử mực nước trong bình hiện là $2m$ và hai người điều khiển có hai quan điểm khác nhau. Người thứ nhất thì cho rằng mực nước như vậy là *đủ* và do đó phải *đóng* van trong khi người thứ hai thì lại cho rằng mực nước $2m$ là *thấp ít* nên phải *mở nhỏ* van.

Nhằm thống nhất hai quan điểm trái ngược nhau đó, ta sẽ đưa thêm vào giá trị độ cao $2m$ một số thực trong khoảng từ 0 đến 1 để đánh giá mức độ phụ thuộc của nó vào hai quan điểm trên. Chẳng hạn độ cao mực nước $2m$ sẽ là *đủ* với độ phụ thuộc 0,7 và *thấp ít* với độ phụ thuộc 0,4. Nếu cả hai người cùng thống nhất với nhau rằng mực nước $2m$ không thể gọi là *thấp nhiều* hoặc *cao* thì mức độ phụ thuộc của nó vào giá trị *thấp nhiều* cũng như vào giá trị *cao* sẽ bằng 0.

Một cách tổng quát thì ta phải đưa thêm vào cho mỗi một độ cao x bất kỳ một số thực $\mu(x)$ trong khoảng [0,1] để đánh giá độ phụ thuộc của nó ứng với từng giá trị ngôn ngữ. Việc đưa thêm số thực $\mu(x)$ để đánh giá độ phụ thuộc như vậy được gọi là *mờ hóa* giá trị rõ x. Ta đi đến định nghĩa:

Định nghĩa: Tập mờ là một tập hợp mà mỗi phần tử cơ bản x của nó được gán thêm một giá trị thực $\mu(x)\in[0,1]$ để chỉ thị *độ phụ thuộc* của phần tử đó vào tập đã cho. Khi độ phụ thuộc bằng 0 thì phần tử cơ bản đó sẽ hoàn toàn không thuộc tập đã cho, ngược lại với độ phụ thuộc bằng 1, phần tử cơ bản sẽ thuộc tập hợp với xác suất 100%.

Như vậy, tập mờ là tập hợp của các cặp $(x, \mu(x))$. Tập kinh điển U của các phần tử x được gọi là *tập nền* của tập mờ. Cho x chạy khắp trong tập hợp U, ta sẽ có hàm $\mu(x)$ có giá trị là số bất kỳ trong khoảng $[0,1]$, tức là

$$\mu : U \rightarrow [0,1]$$

và hàm này được gọi là *hàm thuộc*.

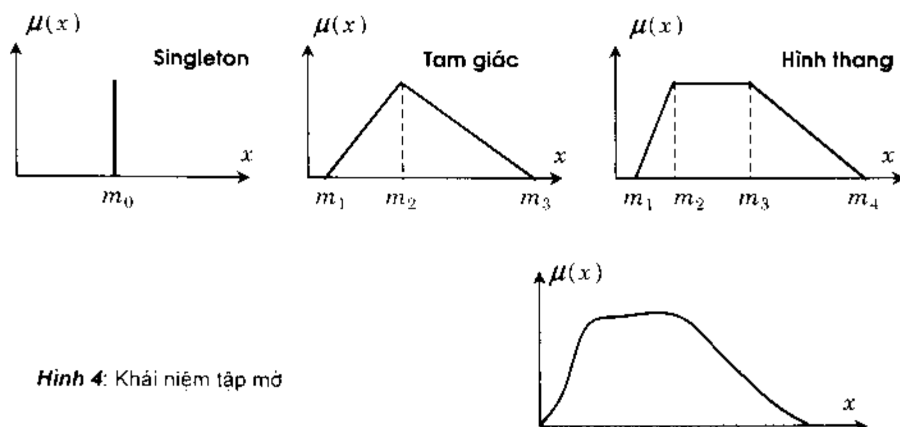
Việc $\mu(x)$ có giá trị là số bất kỳ trong khoảng $[0,1]$ là điều khác biệt cơ bản giữa tập kinh điển và tập mờ. Ở tập kinh điển A, hàm thuộc $\mu(x)$ chỉ có hai giá trị:

$$\mu(x) = \begin{cases} 1 & \text{nếu} \quad x \in A \\ 0 & \text{nếu} \quad x \notin A \end{cases} \tag{4}$$

Chính do có sự khác biệt đó mà ta cũng có nhiều công thức khác nhau cùng mô tả cho một phép tính giữa các tập mờ. Đó là những công thức có cùng một giá trị nếu hàm thuộc $\mu(x)$ thỏa mãn (4).

Như đã nói, bất cứ một hàm $\mu : U \rightarrow [0,1]$ cũng đều có thể là hàm thuộc của một tập mờ nào đó. Song trong điều khiển, với mục đích sử dụng các hàm thuộc sao cho khả năng tích hợp chúng là đơn giản, người ta thường chỉ quan tâm đến ba dạng (hình 4):

- Hàm Singleton (còn gọi là hàm Kronecker).
- Hàm hình tam giác.
- Hàm hình thang.



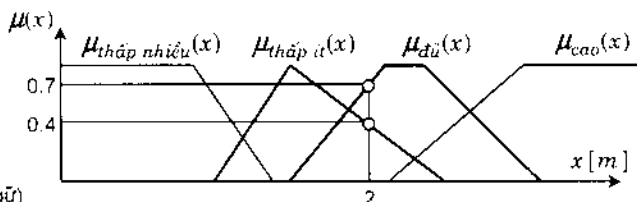
Hình 4: Khái niệm tập mờ

Ví dụ: Thông thường, để chỉ một tập mờ người ta hay sử dụng ngay hàm thuộc $\mu(x)$ của tập mờ đó. Với việc đưa khái niệm tập mờ, mỗi một giá trị ngôn ngữ sẽ là một tập mờ. Trong ví dụ về điều khiển mực nước, ta sẽ có tất cả là bốn tập mờ cho bốn giá trị ngôn ngữ đầu vào:

- Tập mờ $\mu_{thấp\ nhiều}(x)$ cho giá trị thấp nhiều.
- Tập mờ $\mu_{thấp\ ít}(x)$ cho giá trị thấp ít.
- Tập mờ $\mu_{đủ}(x)$ cho giá trị đủ.
- Tập mờ $\mu_{cao}(x)$ cho giá trị cao.

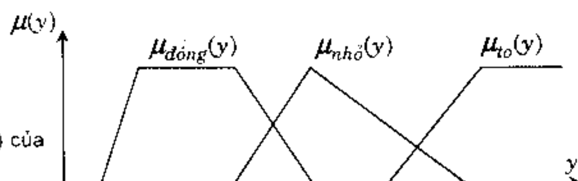
và do đó khi x=2m thì (hình 5)

$$\mu_{thấp\ nhiều}(x) = 0, \quad \mu_{cao}(x) = 0, \quad \mu_{thấp\ ít}(x) = 0.4 \quad \text{và} \quad \mu_{đủ}(x) = 0.7.$$



Hình 5: Các giá trị mờ (ngôn ngữ) của biến vào.

Tương tự, ứng với ba giá trị ngôn ngữ đầu ra *to*, *nhỏ*, *đóng* của biến *van* ta cũng có ba tập mờ $\mu_{to}(y)$, $\mu_{nhỏ}(y)$ và $\mu_{đóng}(y)$ như hình 6 mô tả.



Hình 6: Các giá trị mờ (ngôn ngữ) của biến ra.

2.2 Phép suy diễn mờ

2.2.1 Xác định giá trị của mệnh đề hợp thành

Sau khi đã mờ hóa giá trị rõ x thông qua tập mờ $\mu(x)$ thì bước tiếp theo là ta phải thực hiện những nguyên tắc điều khiển đã cho dưới dạng *mệnh đề hợp thành*. Chẳng hạn ở bài toán điều khiển mực nước là việc thực hiện bốn nguyên tắc:

a) Nếu *mực nước* = *thấp nhiều* thì *van* = *to*.
b) Nếu *mực nước* = *thấp ít* thì *van* = *nhỏ*.
c) Nếu *mực nước* = *cao* thì *van* = *đóng*.
d) Nếu *mực nước* = *đủ* thì *van* = *đóng*.

Chúng đều có chung một cấu trúc đơn:

Nếu $\mathcal{A}=A$ **thì** $\mathcal{B}=B$. (5)

Gọi tập mờ của giá trị A là $\mu_A(x)$ và của B là $\mu_B(y)$. Vậy thì mệnh đề hợp thành (5) sẽ chính là phép suy diễn:

$$A \Rightarrow B \qquad \text{hay} \qquad \mu_A(x) \Rightarrow \mu_B(y) \tag{6}$$

Phép suy diễn (6) là một phép tính có đối số x nên nó cũng phải có một giá trị cụ thể khi mà đối số x, tức là $\mu_A(x)$ đã cho trước. Ký hiệu giá trị của phép suy diễn là $\mu_{A\Rightarrow B}(y)$ thì ở tập kinh điển, nó sẽ được tính từ $\mu_A(x)$, $\mu_B(y)$ như sau:

$$\mu_{A\Rightarrow B}(y) = \mu_A(x)\cdot\mu_B(y) \tag{7a}$$

hoặc

$$\mu_{A\Rightarrow B}(y) = \min\{\mu_A(x)\ ,\ \mu_B(y)\} \tag{7b}$$

Sở dĩ cả hai công thức trên cùng được sử dụng cho tập kinh điển mà không gây mâu thuẫn là vì với x, y thỏa mãn (4), cả hai công thức đó đều cho cùng một giá trị, nói cách khác chúng là tương đương.

Tập kinh điển	Tập mờ
$\mu_A(x)\cdot\mu_B(y) = \min\{\mu_A(x)\ ,\ \mu_B(y)\}$	$\mu_A(x)\cdot\mu_B(y) \neq \min\{\mu_A(x)\ ,\ \mu_B(y)\}$

Với tập mờ $\mu_A(x)$, $\mu_B(y)$ thì điều đó có khác một chút. Hai công thức trên sẽ cho hai giá trị mờ có cùng nền với tập mờ B nhưng với hai hàm thuộc khác nhau. Vậy thì phải bỏ công thức nào ?. Câu trả lời thật khó, và vì khó trả lời như vậy nên người ta đã đề nghị là không bỏ và có thể chọn một trong hai công thức trên, còn chọn như thế nào là do người thiết kế bộ điều khiển tự quyết định:

- Nếu chọn công thức (7a) thì ta nói phép suy diễn mờ đó là *luật suy diễn Prod*.
- Ngược lại nếu chọn (7b) thì phép suy diễn mờ có tên là *luật suy diễn Min*.

Sau khi đã chọn được một công thức thực hiện phép suy diễn là Prod hay Min thì khi cho trước giá trị rõ x_0 ở đầu vào ta luôn có được một giá trị cho phép suy diễn $A \Rightarrow B$. Giá trị đó là tập mờ có hàm thuộc $\mu_{A\Rightarrow B}(y)$ cùng nền với B và được tính như sau (hình 7):

1) $\mu_{A\Rightarrow B}(y) = H\cdot\mu_B(y)$ nếu chọn luật *Prod* (8a)

2) $\mu_{A\Rightarrow B}(y) = \min\{H,\ \mu_B(y)\}$ nếu chọn luật *Min* (8b)

trong đó

$$H = \mu_A(x_0)$$

được gọi là *độ thỏa mãn đầu vào*.

Ví dụ 1: Quay lại bài toán điều khiển mực nước với 4 quy tắc điều khiển có dạng của phép suy diễn:

R_1: Nếu *mực nước = thấp nhiều* thì *van = to*.

R_2: Nếu *mực nước = thấp ít* thì *van = nhỏ*.

R_3: Nếu *mực nước = cao* thì *van = đóng*.

R_4: Nếu *mực nước = đủ* thì *van = đóng*.

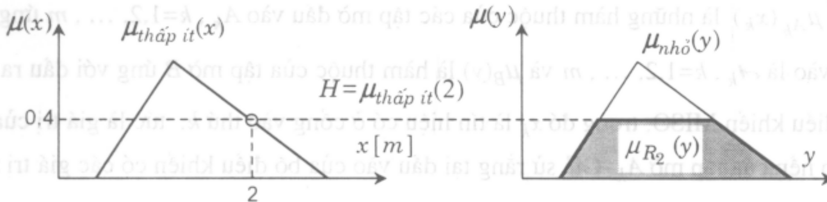
Giả sử rằng mực nước hiện thời là $2m$ và luật thực hiện phép suy diễn được sử dụng là luật *Min* với công thức (8b). Vậy thì:

1) Phép suy diễn (mệnh đề hợp thành) R_1 có giá trị là

$$\mu_{R_1}(y) = \mu_{thấp\ nhiều \Rightarrow to}(y) = \min\{\mu_{thấp\ nhiều}(2), \mu_{to}(y)\} = 0.$$

2) Phép suy diễn (mệnh đề hợp thành) R_2 có giá trị là (hình 7)

$$\mu_{R_2}(y) = \mu_{thấp\ ít \Rightarrow nhỏ}(y) = \min\{\mu_{thấp\ ít}(2), \mu_{nhỏ}(y)\} = \min\{0{,}4\ ,\ \mu_{nhỏ}(y)\}.$$



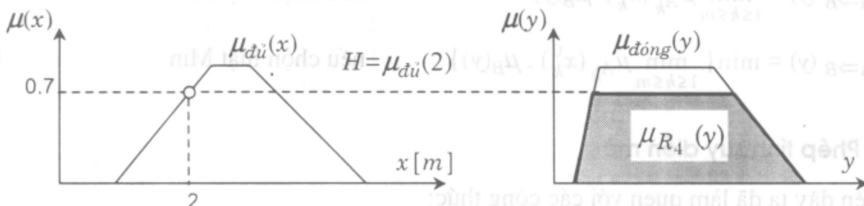
Hình 7: Xác định giá trị của phép suy diễn (mệnh đề hợp thành) R_2 ứng với giá trị rõ x_0=$2m$ tại đầu vào.

3) Phép suy diễn (mệnh đề hợp thành) R_3 có giá trị là

$$\mu_{R_3}(y) = \mu_{cao \Rightarrow đóng}(y) = \min\{\mu_{cao}(2), \mu_{đóng}(y)\} = 0.$$

4) Phép suy diễn (mệnh đề hợp thành) R_4 có giá trị là (hình 8)

$$\mu_{R_4}(y) = \mu_{đủ \Rightarrow đóng}(y) = \min\{\mu_{đủ}(2), \mu_{đóng}(y)\} = \min\{0{,}7\ ,\ \mu_{đóng}(y)\} = 0.$$



Hình 8: Xác định giá trị của phép suy diễn (mệnh đề hợp thành) R_4 ứng với giá trị rõ x_0=$2m$ tại đầu vào.

Trên đây là những bước tính thực hiện việc xác định giá trị mờ $\mu_{A\Rightarrow B}(y)$ của phép suy diễn (6) để thực hiện việc cài đặt mệnh đề hợp thành đơn (5) cho một giá trị rõ x_0 đã biết trước của tín hiệu đầu vào. Bước tiếp theo, ta sẽ nghiên cứu việc thực hiện một mệnh đề MISO:

$$\textbf{Nếu } \mathcal{A}_1 = A_1 \textbf{ và } \cdots \textbf{ và } \mathcal{A}_m = A_m \textbf{ thì } \mathcal{B} = B \tag{9}$$

của bộ điều khiển có nhiều tín hiệu vào và một tín hiệu ra.

So sánh (5) với (9) ta thấy ở mệnh đề hợp thành MISO (9) có nhiều tập mờ đầu vào còn ở mệnh đề (5) chỉ có một đầu vào. Điều này làm cho ta chưa thể sử dụng được ngay một trong hai công thức suy diễn (8a) hoặc (8b) để xác định giá trị mờ $\mu_{A\Rightarrow B}(y)$ vì chưa có được một độ thỏa mãn đầu vào H cụ thể. Nói cách khác, trước khi sử dụng hai công thức suy diễn (8a) hoặc (8b) cho mệnh đề hợp thành (9) ta phải có được độ thỏa mãn đầu vào H chung làm đại diện cho tất cả m các giá trị tín hiệu vào.

Gọi $\mu_{A_k}(x_k)$ là những hàm thuộc của các tập mờ đầu vào A_k, $k=1,2,\ldots,m$ ứng với m tín hiệu vào là $\mathcal{A}_k$, $k=1,2,\ldots,m$ và $\mu_B(y)$ là hàm thuộc của tập mờ B ứng với đầu ra $\mathcal{B}$ của một bộ điều khiển MISO, trong đó x_k là tín hiệu có ở cổng vào thứ k, tức là giá trị của nó sẽ thuộc tập nền của tập mờ A_k. Giả sử rằng tại đầu vào của bộ điều khiển có các giá trị rõ x_k^0, $k=1,2,\ldots,m$. Vậy thì mỗi một tập mờ A_k sẽ có một độ thỏa mãn riêng

$$H_k = \mu_{A_k}(x_k^0).$$

Độ thỏa mãn đầu vào chung H cho cả mệnh đề hợp thành MISO (9) khi đó sẽ được xác định theo nguyên tắc *tình huống xấu nhất* như sau:

$$H = \min\{H_1, H_2, \ldots, H_m\} = \min_{1\le k\le m} \mu_{A_k}(x_k^0).$$

Khi đã có độ thỏa mãn đầu vào chung H thì tập mờ $\mu_{A\Rightarrow B}(y)$ của mệnh đề (9) ứng với vector các giá trị rõ đầu vào x_k^0, $k=1,2,\ldots,m$ sẽ được tính theo công thức (8a) hoặc (8b), tức là:

5) $\mu_{A\Rightarrow B}(y) = \min_{1\le k\le m} \mu_{A_k}(x_k^0)\cdot\mu_B(y)$ nếu chọn luật Prod (10a)

6) $\mu_{A\Rightarrow B}(y) = \min\{\min_{1\le k\le m} \mu_{A_k}(x_k^0),\ \mu_B(y)\}$ nếu chọn luật Min (10b)

2.2.2 Phép tính suy diễn mờ

Trên đây ta đã làm quen với các công thức:

- (8a) và (8b) cho mệnh đề hợp thành SISO,
- (10a) và (10b) cho mệnh đề hợp thành MISO.

phục vụ việc xác định kết quả của mệnh đề hợp thành (phép suy diễn).

Không bị bó buộc bởi chỉ có các công thức đó, thì một cách tổng quát về phép tính suy diễn, mọi ánh xạ $\mu_{A \Rightarrow B} : [0,1]^2 \rightarrow [0,1]$, nếu thỏa mãn:

a) $\mu_{A \Rightarrow B}(H, \mu_B) \leq H$ với mọi $H, \mu_B \in [0,1]$,

b) $\mu_{A \Rightarrow B}(H, \mu_B) \leq \mu_{A \Rightarrow C}(H, \mu_C)$ với mọi $\mu_B \leq \mu_C$ và $H \in [0,1]$,

c) $\mu_{A \Rightarrow B}(H_1, \mu_B) \leq \mu_{A \Rightarrow C}(H_2, \mu_B)$ với mọi $H_1 \leq H_2$ và $\mu_B \in [0,1]$,

d) $\mu_{A \Rightarrow B}(0, \mu_B) = 0$ với mọi $\mu_B \in [0,1]$,

e) $\mu_{A \Rightarrow B}(1, 0) = 0$,

đều có thể được sử dụng để làm hàm thuộc mô tả cho phép tính suy diễn.

2.3 Phép hợp mờ

Tại sao điều khiển lại phải cần đến phép hợp của hai hay nhiều tập mờ ?. Câu trả lời là vì chỉ khi đó ta mới xác định được giá trị chung của một luật hợp thành gồm có nhiều mệnh đề hợp thành.

2.3.1 Xác định giá trị của luật hợp thành

Xét luật hợp thành gồm n mệnh đề hợp thành:

R_1: **Nếu** $\mathcal{A}_1 = A_{11}$ **và** ⋯ **và** $\mathcal{A}_m = A_{1m}$ **thì** $\mathcal{B} = B_1$ **hoặc**

R_2: **Nếu** $\mathcal{A}_1 = A_{21}$ **và** ⋯ **và** $\mathcal{A}_m = A_{2m}$ **thì** $\mathcal{B} = B_2$ **hoặc**

⋮

R_n: **Nếu** $\mathcal{A}_1 = A_{n1}$ **và** ⋯ **và** $\mathcal{A}_m = A_{mn}$ **thì** $\mathcal{B} = B_n$.

Nếu vector các giá trị rõ đầu vào x_0^k, $k=1,2, \ldots, m$ là đã biết trước thì theo công thức (10a) hoặc (10b), mỗi một mệnh đề hợp thành trong luật hợp thành trên sẽ có một giá trị là một tập mờ R_i với hàm thuộc $\mu_{R_i}(y) = \mu_{A_i \Rightarrow B}(y)$, $i=1,2, \ldots, n$. Vì luật hợp thành đang xét có n mệnh đề hợp thành nên ta cũng có ở đây n tập mờ R_i. Vấn đề đặt ra ở đây là từ n tập mờ R_i, $i=1,2, \ldots, n$ đó ta phải xác định được tập mờ kết quả chung R cho toàn bộ luật hợp thành theo phép tính hợp các tập mờ R_i:

$$R = \bigcup_{i=1}^{n} R_i \, . \tag{11}$$

Lý do cho việc sử dụng phép hợp là vì các mệnh đề hợp thành trong một luật hợp thành được liên kết với nhau bằng toán tử "*hoặc*".

Giống như đã làm với phép suy diễn, để thực hiện công thức (11) cho n tập mờ R_q, ta bắt đầu với tập kinh điển. Cho hai tập hợp kinh điển A và B. Gọi $\mu_A(y)$ và $\mu_B(y)$ là những hàm thuộc của chúng. Vậy thì tập $A \cup B$ là kết quả hợp của hai tập trên sẽ có hàm thuộc

$$\mu_{A\cup B}(y) = \begin{cases} 1 & \text{nếu} \quad y \in A \quad \text{hoặc} \quad y \in B \\ 0 & \text{nếu} \quad y \notin A \quad \text{và} \quad y \notin B \end{cases}.$$

Hàm thuộc $\mu_{A\cup B}(y)$ này có thể được suy ra từ hai hàm thuộc

$$\mu_A(y) = \begin{cases} 1 & \text{nếu} \quad y \in A \\ 0 & \text{nếu} \quad y \notin A \end{cases} \quad \text{và} \quad \mu_B(y) = \begin{cases} 1 & \text{nếu} \quad y \in B \\ 0 & \text{nếu} \quad y \notin B \end{cases}$$

bằng một trong hai công thức:

f) $\mu_{A\cup B}(y) = \max\{\mu_A(y), \mu_B(y)\}$ (12a)

g) $\mu_{A\cup B}(y) = \min\{1, \mu_A(y)+\mu_B(y)\}$ (12b)

vì chúng là tương đương.

Tập kinh điển	Tập mờ
$\max\{\mu_A(y), \mu_B(y)\} = \min\{1, \mu_A(y)+\mu_B(y)\}$	$\max\{\mu_A(y), \mu_B(y)\} \neq \min\{1, \mu_A(y)+\mu_B(y)\}$

Khi A và B không phải là tập hợp kinh điển nữa mà là hai tập mờ thì do các hàm thuộc $\mu_A(x)$ và $\mu_B(y)$ của chúng không còn là hàm hai trị 0 hoặc 1 nên tính tương đương của (12a) và (12b) cũng mất. Người thiết kế bộ điều khiển mờ phải tự quyết định lấy cho mình là nên sử dụng công thức nào:

1) Nếu sử dụng công thức (12a) thì người ta nói phép hợp các tập mờ đã được thực hiện theo *luật Max* (cực đại).
2) Nếu sử dụng công thức (12b) thì người ta nói phép hợp các tập mờ đã được thực hiện theo *luật Sum* (tổng).

Ví dụ 2: Trở lại bài toán ban đầu là điều khiển mực nước với luật điều khiển đã biết

R_1: Nếu *mực nước* = *thấp nhiều* thì *van* = *to*.

R_2: Nếu *mực nước* = *thấp ít* thì *van* = *nhỏ*.

R_3: Nếu *mực nước* = *cao* thì *van* = *đóng*.

R_4: Nếu *mực nước* = *đủ* thì *van* = *đóng*.

Sau khi đã xây dựng được các hàm thuộc (mục 2.1):

a) $\mu_{thấp\ nhiều}(x)$, $\mu_{thấp\ ít}(x)$, $\mu_{đủ}(x)$, $\mu_{cao}(x)$

b) $\mu_{đóng}(y)$, $\mu_{nhỏ}(y)$, $\mu_{to}(y)$

mô tả cho những giá trị ngôn ngữ của các biến vào/ra và nhất là sau khi đã chọn luật Min cho việc thực hiện phép suy diễn thì ứng với mực nước hiện thời $2m$ ta đã có được các kết quả sau của 4 mệnh đề hợp thành trên (mục 2.2.1):

a) Phép suy diễn *thấp nhiều* $\Rightarrow$ *to*: $\mu_{R_1}(y) = 0$.

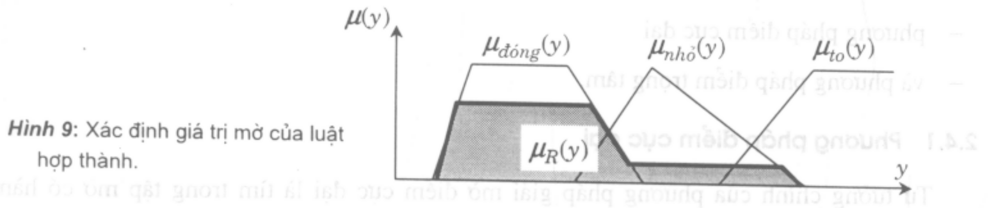
b) Phép suy diễn *thấp ít* $\Rightarrow$ *nhỏ*: $\mu_{R_2}(y) = \min\{0{,}4\ ,\ \mu_{nhỏ}(y)\}$.

c) Phép suy diễn *cao* $\Rightarrow$ *đóng*: $\mu_{R_3}(y) = 0$.

d) Phép suy diễn *đủ* $\Rightarrow$ *đóng*: $\mu_{R_4}(y) = \min\{0{,}7\ ,\ \mu_{đóng}(y)\}$.

Bước tiếp theo là ta phải xác định tập mờ chung của cả 4 tập mờ trên để làm kết quả $\mu_R(y)$ cho luật hợp thành ứng với mực nước đầu vào $2m$. Nếu chọn phép hợp mờ *Max* với công thức tính (12a) thì (hình 9):

$$\mu_R(y) = \max\{0, \mu_{R_2}(y)\ ,\ 0\ ,\ \mu_{R_4}(y)\ \} = \max\{\ \mu_{R_2}(y)\ ,\ \mu_{R_4}(y)\ \}.$$



Hình 9: Xác định giá trị mờ của luật hợp thành.

2.3.2 Phép tính hợp các tập mờ

Mục 2.3.1 đã giới thiệu hai công thức tính hợp của các tập mờ. Một cách tổng quát thì mọi hàm $\mu : [0,1]^2 \to [0,1]$, đều có thể được sử dụng để xác định hàm thuộc cho $A \cup B$ nếu chúng thỏa mãn:

a) $\mu(x,\ y) = \mu(y,\ x)$.

b) $\mu(x,\ y) \le \mu(u,\ v)$ nếu $x \le u$ và $y \le v$,

c) $\mu(x,\ \mu(y,z)) = \mu(\mu(x,\ y),\ z)$,

d) $\mu(0,x) = x$.

trong đó $x,\ y,\ u,\ v,\ z \in [0,1]$.

Ví dụ 3:

1) $\mu_{A\cup B}(y) = \mu_A(x) + \mu_B(y) - \mu_A(x)\mu_B(y)$. (Tổng trực tiếp)

2) $\mu_{A\cup B}(y) = \dfrac{\mu_A(y) + \mu_B(y)}{1 + \mu_A(y) + \mu_B(y)}$. (Tổng Einstein)

3) $\mu_{A\cup B}(y) = \begin{cases} \max\{\mu_A(y), \mu_B(y)\} & \text{khi } \min\{\mu_A(y), \mu_B(y)\} = 0 \\ 1 & \text{khi } \min\{\mu_A(y), \mu_B(y)\} \ne 0 \end{cases}$

Chú ý: Khác với phép suy diễn, phép hợp hai tập mờ chỉ có nghĩa khi chúng có cùng nền.

2.4 Giải mờ

Sau khi thực hiện được xong việc tính giá trị luật hợp thành (nguyên lý điều khiển) chúng ta thu được kết quả là tập mờ $\mu_R(y)$ cùng nền với tín hiệu ra. Kết quả đó chưa thể là một giá trị thích hợp để điều khiển. Chẳng hạn như ở bài toán điều khiển mực nước, tuy rằng đã xác định được kết quả của luật điều khiển là tập mờ có hàm thuộc $\mu_R(y)$ cho mực nước $2m$ như ở hình 9, ta vẫn không biết được phải chỉnh van nước như thế nào, nói cách khác ta vẫn chưa biết phải điều chỉnh van một góc mở là bao nhiêu?

Công việc xác định một góc mở van cụ thể, hay nói một cách tổng quát, việc xác định một giá trị rõ y_0 từ tập mờ $\mu_R(y)$ của nó, được gọi là *giải mờ*. Giá trị rõ y_0 xác định được có thể được xem như "*phần tử đại diện xứng đáng*" cho tập mờ.

Căn cứ theo những quan niệm khác nhau về *phần tử đại diện xứng đáng* mà ta sẽ có các phương pháp giải mờ khác nhau. Trong điều khiển người ta thường hay sử dụng hai phương pháp chính, đó là:

- phương pháp điểm cực đại
- và phương pháp điểm trọng tâm.

2.4.1 Phương pháp điểm cực đại

Tư tưởng chính của phương pháp giải mờ điểm cực đại là tìm trong tập mờ có hàm thuộc $\mu_R(y)$ một phần tử rõ y_0 với độ phụ thuộc lớn nhất (có xác suất thuộc tập mờ lớn nhất trong số những phần tử còn lại), tức là:

$$y_0 = \arg\max_{y} \mu_R(y). \qquad (13)$$

Tuy nhiên, do việc tìm y_0 theo (13) có thể đưa đến vô số nghiệm (hình 10) nên ta phải đưa thêm những yêu cầu cho phép chọn trong số các nghiệm đó một giá trị y_0 cụ thể *chấp nhận được*. Như vậy, việc giải mờ theo phương pháp cực đại sẽ gồm hai bước:

- xác định miền chứa giá trị rõ y_0. Giá trị rõ y_0 là giá trị mà tại đó hàm thuộc đạt giá trị cực đại (bằng độ thỏa mãn đầu vào H), tức là miền

  $$G = \{\, y \in Y \mid \mu_R(y) = H \,\}.$$

- xác định y_0 có thể chấp nhận được từ G.

Trong ví dụ ở hình 10 thì G là khoảng $[y_1, y_2]$ của tập nền của R.

Trong trường hợp có vô số nghiệm của (13) thì để tìm y_0 ta có hai cách:

1) *Xác định điểm trung bình*:

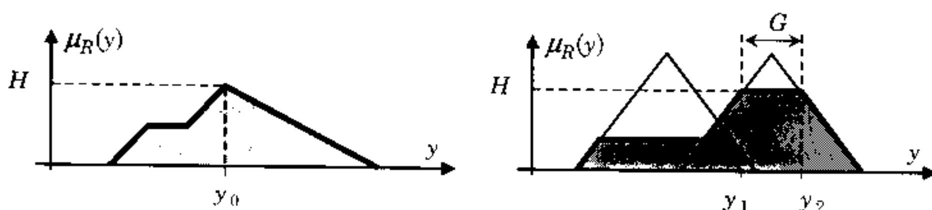
$$y_0 = \frac{y_1 + y_2}{2}.$$

Nếu các hàm thuộc đều có dạng tam giác hoặc hình thang thì điểm y_0 xác định theo phương pháp này sẽ không quá bị nhạy cảm với sự thay đổi của giá trị rõ đầu vào x_0 do đó rất thích hợp với các bài toán có nhiễu biên độ nhỏ tại đầu vào.

2) *Xác định điểm cận trái hoặc phải*:

$$y_0 = \inf_{y \in G}(y) \quad \text{hoặc} \quad y_0 = \sup_{y \in G}(y).$$

Theo phương pháp giải mờ này và nếu các hàm thuộc đều có dạng tam giác hoặc hình thang thì điểm y_0 sẽ phụ thuộc tuyến tính (trong một lân cận) vào giá trị rõ x_0 tại đầu vào.



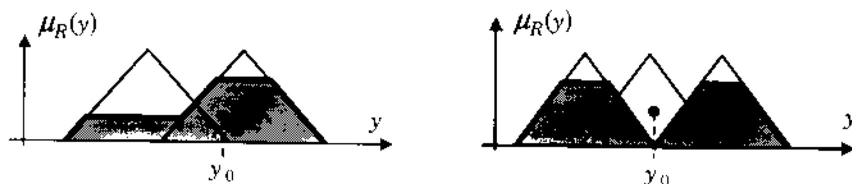
Hình 10 : Giải mờ bằng phương pháp cực đại.

2.4.2 Phương pháp điểm trọng tâm

Phương pháp điểm trọng tâm sẽ cho ra kết quả y_0 là hoành độ của điểm trọng tâm miền được bao bởi trục hoành và đường $\mu_R(y)$ – hình 11.

$$y_0 = \frac{\int_S y\mu_R(y)dy}{\int_S \mu_R(y)dy}. \tag{14}$$

với $S = \text{supp}\mu_R(y) = \{y \mid \mu_R(y) \neq 0\}$ là miền xác định của tập mờ R.



Hình 11 : Giải mờ bằng phương pháp điểm trọng tâm.

Đây là phươngg pháp ưa được sử dụng nhất. Nó cho phép ta xác định giá trị y_0 với sự tham gia của tất cả các tập mờ đầu ra của luật điều khiển một cách bình đẳng và chính xác. Tuy nhiên phương pháp này lại không để ý được tới độ thỏa mãn của mệnh đề điều khiển cũng như thời gian tính lâu. Ngoài ra một trong những nhược điểm cơ bản của phương pháp

điểm trọng tâm là có thể giá trị y_0 xác định được lại có độ thuộc nhỏ nhất, thậm chí bằng 0 (hình 11 bên phải là một ví dụ minh họa).

Phương pháp điểm trọng tâm cho thiết bị hợp thành Sum-Min

Xét một luật hợp thành có n mệnh đề. Nếu tập mờ đầu ra $\mu_R(y)$ của nó được xác định với:

- luật suy diễn *Min* (công thức (8b) hoặc (10b)) và,
- luật hợp *Sum* (công thức (12b)),

thì người ta nói $\mu_R(y)$ đã được tính theo quy tắc *Sum-Min*. Một luật hợp thành có kèm theo quy tắc tính *Sum-Min* được gọi là *thiết bị hợp thành Sum–Min*.

Với quy tắc *Sum-Min*, và nếu không cần phải để ý tới giá trị cực đại của hàm thuộc không được lớn hơn 1 (vẫn được áp dụng trong thực tế ứng dụng của điều khiển mờ) thì giá trị mờ $\mu_R(y)$ đầu ra của luật hợp thành sẽ chính là tổng của n giá trị mờ đầu ra $\mu_{R_i}(y)$, $i=1,\cdots,n$ của từng mệnh đề hợp thành.

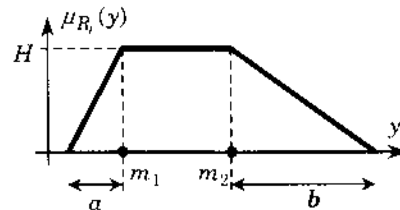
$$\mu_R(y) = \sum_{i=1}^{q} \mu_{R_i}(y). \tag{15}$$

Thay (15) vào (14), sau đó đổi chỗ của tổng và tích phân cho nhau (hoàn toàn có nghĩa, vì tổng và tích phân đều hội tụ) thì công thức tính y_0 sẽ được đơn giản như sau

$$y_0 = \frac{\int\limits_S \left(y \sum\limits_{i=1}^{n} \mu_{R_i}(y) \right) dy}{\int\limits_S \sum\limits_{i=1}^{n} \mu_{R_i}(y) dy} = \frac{\sum\limits_{i=1}^{n} \left(\int\limits_S y \mu_{R_i}(y) dy \right)}{\sum\limits_{i=1}^{n} \left(\int\limits_S \mu_{R_i}(y) dy \right)} = \frac{\sum\limits_{i=1}^{n} M_i}{\sum\limits_{i=1}^{n} A_i},$$

trong đó

$$M_i = \int\limits_S y \mu_{R_i}(y) dy \quad \text{và} \quad A_i = \int\limits_S \mu_{R_i}(y) dy, \qquad i=1,\cdots,n.$$



Hình 12: Tập mờ có hàm thuộc hình thang.

Xét riêng cho các hàm thuộc $\mu_{R_i}(y)$ dạng hình thang như trong hình 12 thì

$$M_k = \frac{H}{6}\left(3m_2^2 - 3m_1^2 + b^2 - a^2 + 3m_2 b + 3m_1 a\right). \quad (16a)$$

$$A_k = \frac{H}{2}(2m_2 - 2m_1 + a + b). \quad (16b)$$

Có một điều đặc biệt ở đây bạn đọc nên chú ý là tuy công thức (16a) được dẫn ra với giả thiết rằng phép tính thực hiện luật hợp thành là *Sum–Min* song trong thực tế nó vẫn được áp dụng ngay cả khi luật hợp thành được thực hiện theo *Max–Min* (thực hiện hợp mờ theo luật *Max* và suy diễn mờ theo luật *Min*).

Ví dụ 4: Quay lại bài toán điều khiển mực nước. Khi đầu vào có giá trị rõ là $2m$ ta đã tính ra được giá trị mờ của luật hợp thành (nguyên tắc điều khiển) theo *Max–Min* với hàm thuộc:

$$\mu_R(y) = \max\{\ \mu_{R_2}(y)\ ,\ \mu_{R_4}(y)\ \}$$

cho trong hình 9. Hình 13 biểu diễn lại hàm thuộc đó một cách chi tiết hơn với các giá trị cụ thể của nó.

Như đã nói, tuy rằng $\mu_R(y)$ được xác định theo luật *Max–Min* nhưng để giải mờ với phương pháp điểm trọng tâm người ta vẫn thường sử dụng công thức (16). Từ hình 13 ta suy ra được

a) Với hình thang $\mu_{R_4}(y)$

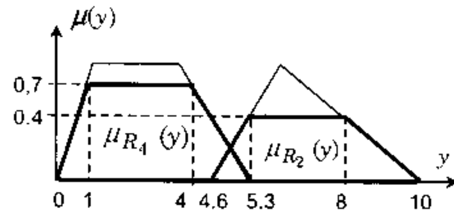
$$M_4 = \frac{0{,}7}{6}\,(3\cdot 4^2 - 3\cdot 1^2 + (5{,}3-4)^2 - (1-0)^2 + 3\cdot 4\cdot(5{,}3-4) + 3\cdot 1\cdot(1-0)) = 7{,}5.$$

$$A_4 = \frac{0{,}7}{2}\,(2\cdot 4 - 2\cdot 1 + (1-0) + (5{,}3-4)) = 2{,}9.$$

b) Với hình thang $\mu_{R_2}(y)$

$$M_2 = \frac{0{,}7}{6}\,(3\cdot 8^2 - 3\cdot 5{,}3^2 + (10-8)^2 - (5{,}3-4{,}6)^2 + 3\cdot 8\cdot(10-8) + 3\cdot 5{,}3\cdot(5{,}3-4{,}6)) = 19{,}9.$$

$$A_2 = \frac{0{,}7}{2}\,(2\cdot 8 - 2\cdot 5{,}3 + (5{,}3-4{,}6) + (10-8)) = 2{,}8.$$



Hình 13: Xác định giá trị rõ của tập mờ.

Vậy

$$y_0 = \frac{M_4 + M_2}{A_4 + A_2} = \frac{7{,}5 + 19{,}9}{2.9 + 2.8} = 4{,}8.$$

Phương pháp độ cao

Đặc biệt hơn nữa ở phương pháp điểm trọng tâm, là nếu các hàm thuộc của tập mờ đầu ra lại có dạng singleton với

$$\mu_{B_i}(y) = \begin{cases} 1 & \text{nếu} \quad y = y_i \\ 0 & \text{nếu} \quad y \neq y_i \end{cases}$$

thì do

$$\mu_{R_i}(y) = \begin{cases} H_i & \text{nếu} \quad y = y_i \\ 0 & \text{nếu} \quad y \neq y_i \end{cases}$$

ta sẽ có

$$y_0 = \frac{\sum_{i=1}^{n} y_i H_i}{\sum_{i=1}^{n} H_i}. \tag{17}$$

Công thức (17) có tên gọi là công thức tính xấp xỉ y_0 theo *phương pháp độ cao* và nó cũng thường được sử dụng không chỉ riêng khi tính giá trị luật hợp thành theo quy tắc *Sum-Min*, mà còn cho cả những quy tắc khác như: *Max–Min*, *Sum–Prod*, *Sum–Min*, ...).

3 Bộ điều khiển mờ

3.1 Cấu trúc một bộ điều khiển mờ

Do bản chất là một bộ điều khiển thực hiện luật hợp thành

R_1: **Nếu $\mathcal{A}_1$=A_{11} và ··· và $\mathcal{A}_m$=A_{m1} thì $\mathcal{B}$=B_1** hoặc

R_2: **Nếu $\mathcal{A}_1$=A_{11} và ··· và $\mathcal{A}_m$=A_{2m} thì $\mathcal{B}$=B_2** hoặc

⋮

R_n: **Nếu $\mathcal{A}_1$=A_{n1} và ··· và $\mathcal{A}_m$=A_{nm} thì $\mathcal{B}$=B_n.**

nên bộ điều khiển mờ phải có ba khâu cơ bản gồm (hình 14):

- Khâu mờ hóa có nhiệm vụ chuyển đổi một giá trị rõ đầu vào

  $\underline{u}_0 = (u_k^0, k=1.2. \ldots . m)$,

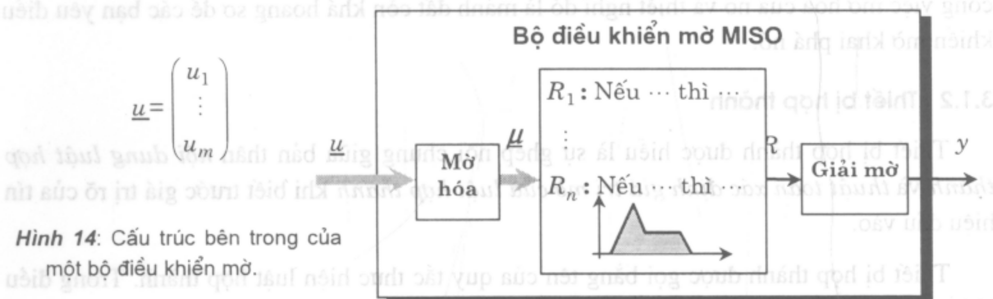
  trong đó ký hiệu u_k^0 không có nghĩa là lũy thừa 0 của x mà đó chỉ là ký hiệu chỉ rằng nó là giá trị rõ của tín hiệu đầu vào thứ k, thành vector

  $\underline{\mu}_i = (\mu_{A_{ki}}(u_k^0), k=1,2, \ldots, m)$

  cho mệnh đề hợp thành thứ i (i=1,2, ... , n) , tức là giá trị rõ ứng với tập mờ A_{ki}.

– Khâu thực hiện luật hợp thành, có tên gọi là *thiết bị hợp thành*, xử lý các vector $\underline{\mu}_i$, i=1,2, … , n và cho ra giá trị mờ R với hàm thuộc $\mu_R(y)$ của biến ngôn ngữ đầu ra.

– Khâu giải mờ, có nhiệm vụ chuyển đổi tập mờ $\mu_R(y)$ thành một giá trị rõ y_0 chấp nhận được cho đối tượng (*tín hiệu điều chỉnh*).



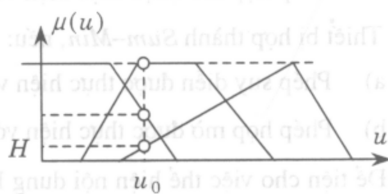
Hình 14: Cấu trúc bên trong của một bộ điều khiển mờ.

3.1.1 Mờ hóa

Như đã giới thiệu, mờ hóa là sự chuyển đổi giá trị rõ đầu vào $\underline{u}_0 = (u_k^0, k=1,2, \dots, m)$, thành vector $\underline{\mu}_i=(\mu_{A_{ki}}(u_k^0)$, để trên cơ sở đó xác định được độ thỏa mãn đầu vào $H_i = \min\limits_{1\le k\le m} \mu_{A_{ki}}(u_k^0)$ phục vụ cho công tác suy diễn (hình 15):

R_i: **Nếu** $\mathcal{U}_1=A_{i1}$ **và** ⋯ **và** $\mathcal{U}_m=A_{im}$ **thì** $\mathcal{Y}=Y_i$.

Đơn giản là vậy, song kinh nghiệm cho thấy đây là một bước khó khăn nhất khi phải thiết kế một bộ điều khiển mờ. Lý do là vì để mờ hóa ta phải có được các hàm thuộc biểu diễn giá trị ngôn ngữ cho biến đầu vào mà điều này lại không thuộc hẳn về nhiệm vụ nghiên cứu của lý thuyết tập mờ lại cũng không hoàn toàn có được trên cơ sở kinh nghiệm điều khiển (ý kiến của chuyên gia). Mặc dù lý thuyết mờ phát triển nhiều, đa dạng, đưa được đến những thuật toán xác định giá trị luật hợp thành cũng như chỉnh định các mệnh đề hợp thành (chẳng hạn thông qua các trọng số) rất hiệu quả, song những kết quả này lại dựa vào giả thiết là đã phải có tập mờ (hàm thuộc). Chuyên gia thì nhiều nhất cũng chỉ có thể cung cấp được ý kiến về miền xác định cho các tập mờ. Vì công việc mờ hóa nằm giữa chừng như vậy nên việc thiết lập các hàm thuộc cho tập mờ hoàn toàn do người thiết kế đảm nhận.



Hình 15: Xác định độ thỏa mãn đầu vào.

Bản thân tác giả bài này cũng gặp rất nhiều khó khăn khi phải giải quyết bước mờ hóa. Thậm chí, cho dù đã "tình cờ" chọn được các hàm thuộc một cách có hiệu quả cho công việc điều khiển thì cũng vẫn không giải thích được một cách cặn kẽ lý do tại sao. Bởi vậy nếu sau này có muốn nâng cao chất lượng của bộ điều khiển thông qua sửa đổi hàm thuộc thì không biết phải bắt đầu từ điểm xuất phát nào.

Tóm lại, chuyên ngành điều khiển mờ còn thiếu nhiều sự hỗ trợ từ phía lý thuyết cho công việc mờ hóa của nó và thiết nghĩ đó là mảnh đất còn khá hoang sơ để các bạn yêu điều khiển mờ khai phá nó.

3.1.2 Thiết bị hợp thành

Thiết bị hợp thành được hiểu là sự ghép nối chung giữa bản thân *nội dung luật hợp thành* và *thuật toán xác định giá trị mờ của luật hợp thành* khi biết trước giá trị rõ của tín hiệu đầu vào.

Thiết bị hợp thành được gọi bằng tên của quy tắc thực hiện luật hợp thành. Trong điều khiển ta có 4 thiết bị chính. Đó là:

1) Thiết bị hợp thành *Max–Min*, nếu:
   a) Phép suy diễn được thực hiện với luật *Min*: $\mu_{A\Rightarrow B}(y) = \min\{H, \mu_B(y)\}$.
   b) Phép hợp mờ được thực hiện theo luật *Max*: $\mu_{A\cup B}(y) = \max\{\mu_A(y), \mu_B(y)\}$.
2) Thiết bị hợp thành *Max–Prod*, nếu:
   a) Phép suy diễn được thực hiện với luật *Prod*: $\mu_{A\Rightarrow B}(y) = H\cdot\mu_B(y)$.
   b) Phép hợp mờ được thực hiện theo luật *Max*: $\mu_{A\cup B}(y) = \max\{\mu_A(y), \mu_B(y)\}$.
3) Thiết bị hợp thành *Sum–Prod*, nếu:
   a) Phép suy diễn được thực hiện với luật *Prod*: $\mu_{A\Rightarrow B}(y) = H\cdot\mu_B(y)$.
   b) Phép hợp mờ được thực hiện với luật *Sum*: $\mu_{A\cup B}(y) = \min\{1, \mu_A(y)+\mu_B(y)\}$.
4) Thiết bị hợp thành *Sum–Min*, nếu:
   a) Phép suy diễn được thực hiện với luật *Min*: $\mu_{A\Rightarrow B}(y) = \min\{H, \mu_B(y)\}$.
   b) Phép hợp mờ được thực hiện với luật *Sum*: $\mu_{A\cup B}(y) = \min\{1, \mu_A(y)+\mu_B(y)\}$.

Để tiện cho việc thể hiện nội dung luật hợp thành trong thiết bị, người ta thường không biểu diễn luật hợp thành dưới dạng các câu văn như ta đã biết mà thay vào đó là bảng, rất thích hợp khi cài đặt với kiểu cấu trúc dữ liệu dạng mảng (array). Chẳng hạn như thay vì:

R_1: Nếu *mực nước* = *thấp nhiều* thì *van* = *to*, hoặc
R_2: Nếu *mực nước* = *thấp ít* thì *van* = *nhỏ*, hoặc
R_3: Nếu *mực nước* = *cao* thì *van* = *đóng*, hoặc
R_4: Nếu *mực nước* = *đủ* thì *van* = *đóng*,

người ta lại biểu diễn thành:

van \ mực nước	thấp nhiều	thấp ít	cao	đủ
to				
nhỏ				
đóng				

Theo cách biểu diễn luật hợp thành dưới dạng bảng như vậy, bảng của một luật hợp thành MISO với m đầu vào, một đầu ra sẽ có m+1 chiều. Để giảm số chiều xuống còn m người ta sử dụng luôn các ô trong bảng biểu diễn giá trị ngôn ngữ cho tín hiệu ra. Ví dụ:

R_1 : Nếu u_1=ZE và u_2=NB thì y=NB hoặc,

R_2 : Nếu u_1=PS và u_2=NB thì y=NS hoặc,

R_3 : Nếu u_1=NS và u_2=NS thì y=NS hoặc,

R_4 : Nếu u_1=ZE và u_2=NS thì y=NS hoặc,

R_5 : Nếu u_1=PS và u_2=NS thì y=NS hoặc,

R_6 : Nếu u_1=PB và u_2=NS thì y=ZE hoặc,

R_7 : Nếu u_1=NB và u_2=ZE thì y=NB hoặc,

R_8 : Nếu u_1=NS và u_2=ZE thì y=NS hoặc,

R_9 : Nếu u_1=ZE và u_2=ZE thì y=ZE hoặc,

R_{10} : Nếu u_1=PS và u_2=ZE thì y=PS hoặc,

R_{11} : Nếu u_1=PB và u_2=ZE thì y=PB hoặc,

R_{12} : Nếu u_1=NS và u_2=PS thì y=PS hoặc,

R_{13} : Nếu u_1=ZE và u_2=PS thì y=PS hoặc,

R_{14} : Nếu u_1=PS và u_2=PS thì y=PS hoặc,

R_{15} : Nếu u_1=PB và u_2=PS thì y=PS hoặc,

R_{16} : Nếu u_1=NS và u_2=PB thì y=PB hoặc,

R_{17} : Nếu u_1=ZE và u_2=PB thì y=PS hoặc,

R_{18} : Nếu u_1=PS và u_2=PB thì y=PB.

sẽ được thể hiện dưới dạng bảng như sau:

u_2 \ u_1	NB	NS	ZE	PS	PB
NB			NB	NS	
NS		NS	NS	NS	ZE
ZE	NB	NS	ZE	PS	PB
PS		PS	PS	PS	PS
PB		PB	PS	PB	

3.1.3 Khâu giải mờ

Đây là thành phần cuối cùng trong bộ điều khiển mờ có nhiệm vụ xác định một phần tử y_0 làm đại diện cho tập mờ R có hàm thuộc $\mu_R(y)$, trong đó $\mu_R(y)$ là kết quả đầu ra của thiết bị hợp thành. Theo như mục 2.4.2 thì người ta thường xác định y_0 như sau:

1) $y_0 = \sup\{y \mid y = \arg\max_{y} \mu_R(y)\}$, phương pháp điểm cực đại bên phải.

2) $y_0 = \inf\{y \mid y = \arg\max_{y} \mu_R(y)\}$, phương pháp điểm cực đại bên trái.

3) $y_0 = \dfrac{\int_S y\mu_R(y)dy}{\int_S \mu_R(y)dy}$, với $S = \text{supp}\mu_R(y)$ là miền xác định của tập mờ R (phương pháp điểm trọng tâm).

3.2 Thiết kế bộ điều khiển mờ

3.2.1 Các bước thực hiện chung

Giả thiết rằng, người thiết kế đã thu thập đủ các kinh nghiệm cũng như ý kiến của chuyên gia và muốn chuyển nó thành bộ điều khiển thì phải tiến hành các bước sau đây:

- Định nghĩa tất cả các biến ngôn ngữ vào và ra, đó cũng chính là các tín hiệu vào/ra của bộ điều khiển.
- Định nghĩa các tập mờ (giá trị ngôn ngữ) cho từng biến vào/ ra, tức là thực hiện công việc mờ hóa.
- Xây dựng luật hợp thành.
- Chọn quy tắc thực hiện luật hợp thành (thiết bị hợp thành), hay còn được gọi là động cơ suy diễn.
- Chọn phương pháp giải mờ.

Trong quá trình thiết kế, ta cần lưu ý mấy điểm sau:

1) Không nên thiết kế bộ điều khiển mờ để giải quyết một bài toán tổng hợp mà có thể dễ dàng thực hiện bằng các bộ điều khiển kinh điển (bộ điều khiển P, - PI, - PD, - PID, bộ điều khiển trạng thái) thoả mãn các yêu cầu đặt ra.
2) Không nên thiết kế bộ điều khiển mờ cho các hệ thống cần độ an toàn cao (điều khiển lò phản ứng hạt nhân, điều khiển các quy trình công nghệ sản xuất hóa chất ...).
3) Do nguyên lý làm việc của bộ điều khiển mờ là sao chép lại kinh nghiệm điều khiển của chuyên gia nên luôn phải nghĩ tới việc bổ sung thêm cho bộ điều khiển mờ các khả năng tự học để thích nghi được với sự thay đổi của đối tượng. Thông thường người ta ít khi yêu cầu một cách khắt khe là hệ thống điều khiển tự động phải có chất lượng cao nhất mà thường là chỉ tiêu bền vững. Tuy rằng sao chép lại nguyên lý điều khiển của chuyên gia, nhưng nếu như đã được chuẩn bị và được tối ưu hoá một cách khéo léo, các bộ điều khiển mờ sẽ có khả năng làm việc bền vững hơn, linh hoạt hơn cả chuyên gia.

3.2.2 Quan hệ truyền đạt

Quan hệ truyền đạt của bộ điều khiển là mô hình toán học mô tả quan hệ $y=f(\underline{u})$ giữa vector các tín hiệu vào $\underline{u}(t)$ và tín hiệu ra $y(t)$. Ở đây chúng tôi đã không gọi quan hệ $y=f(\underline{u})$ của hệ mờ là mô hình vào/ra như trong điều khiển kinh điển vẫn thường gọi mà thay vào đó là khái niệm *quan hệ truyền đạt*. Lý do đơn giản chỉ là để nhấn mạnh rằng chúng ta sẽ không bị bắt buộc phải có mô hình khi thiết kế một bộ điều khiển mờ.

Nếu đã không cần mô hình thì tại sao ta lại đặt ra vấn đề nghiên cứu quan hệ truyền đạt $y=f(\underline{u})$ của hệ mờ?. Đó là để phục vụ việc phân tích đánh giá chất lượng hệ mờ. Hơn nữa đôi khi trong thực tế ta vẫn thường hay gặp phải bài toán thiết kế một bộ điều khiển mờ có $y=f(\underline{u})$ cho trước.

Nhìn lại từng khâu của bộ điều khiển mờ gồm các khâu mờ hóa, thiết bị hợp thành và giải mờ trong hình 15, thì thấy rằng trong quan hệ truyền đạt, giá trị $y_0=y(t_0)$ tại thời điểm $t=t_0$ ở đầu ra chỉ phụ thuộc vào một mình giá trị $\underline{u}_0=\underline{u}(t_0)$ của đầu vào tại đúng thời điểm đó chứ không phụ thuộc vào các giá trị đã qua của tín hiệu $u(t)$, tức là không phụ thuộc vào tích phân hay đạo hàm của $\underline{u}(t)$. Như vậy $y=f(\underline{u})$ là một hàm đại số và do đó bộ điều khiển mờ thực chất là một bộ điều khiển (*phi tuyến*) *tĩnh*.

Quan hệ truyền đạt $f(\underline{u})$ của bộ điều khiển mờ với m đầu vào, một đầu ra (MISO) và luật hợp thành gồm n mệnh đề hợp thành

$$R_i: \quad \textbf{Nếu } \mathcal{U}_1=A_{i1} \textbf{ và } \cdots \textbf{ và } \mathcal{U}_m=A_{im} \textbf{ thì } \mathcal{Y}=Y_i, \quad i=1,2,\ldots,n$$

sẽ nhận được thông qua thực hiện việc ghép nối các ánh xạ:

a) $\underline{u}_0 \mapsto H_i = \min\limits_{1\le k\le m} \mu_{A_{ki}}(u_k^0)$ của khâu mờ hóa, trong đó $\underline{u}_0=(u_k^0, k=1,2,\ldots,m)$ là một giá trị rõ ở đầu vào, A_{ki}, $k=1,2,\ldots,m$, $i=1,2,\ldots,n$ là các tập mờ ứng với n giá trị ngôn ngữ cho từng đầu vào.

b) $H_i \mapsto \mu_{R_i}(y)$ của phép suy diễn.

c) $\mu_{R_i}(y) \mapsto \mu_R(y)$ của phép hợp mờ.

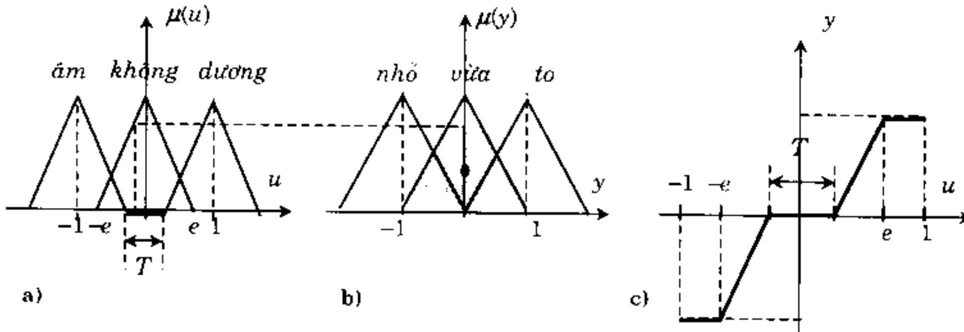
d) $\mu_R(y) \mapsto y_0$ của khâu giải mờ.

Ví dụ 5: Xét bộ điều khiển mờ có luật hợp thành:

R_1:	Nếu $\mathcal{U}$ =*âm* thì $\mathcal{Y}$=*nhỏ*	hoặc,
R_2:	Nếu $\mathcal{U}$ =*không* thì $\mathcal{Y}$=*vừa*	hoặc,
R_3:	Nếu $\mathcal{U}$ =*dương* thì $\mathcal{Y}$=*to*.	

trong đó các giá trị ngôn ngữ *âm*, *không*, *dương* của biến $\mathcal{U}$ và *nhỏ*, *vừa*, *to* của biến $\mathcal{Y}$ cho trong hình 16a) và 16b). Nếu bộ điều khiển mờ được cài đặt với thiết bị hợp thành

Max–Min và giải mờ theo phương pháp điểm trọng tâm thì nó sẽ có quan hệ hợp thành cho trong hình 16c).



Hình 16: Minh họa cho ví dụ 5.

Nếu cài đặt một luật hợp thành gồm n mệnh đề hợp thành với thiết bị hợp thành *Sum–Prod* và giải mờ theo phương pháp điểm trọng tâm, thì bộ điều khiển thu được sẽ có quan hệ truyền đạt:

$$y_0 = \frac{\sum_{i=1}^{n}\left(\int_S y\mu_{R_i}(y)\min_{1\le k\le m}\mu_{A_{ki}}(u_k^0)dy\right)}{\sum_{i=1}^{n}\left(\int_S \mu_{R_i}(y)\min_{1\le k\le m}\mu_{A_{ki}}(u_k^0)dy\right)}.$$

3.2.3 Tổng hợp bộ điều khiển có quan hệ truyền đạt cho trước (13)

Bây giờ ta sẽ xét bài toán tổng hợp một bộ điều khiển mờ SISO khi biết trước quan hệ truyền đạt $y=f(u)$ của nó. Đây là bài toán thường gặp khi mà ta đã áp dụng phương pháp kinh điển để phân tích hệ thống và đã đến được mô hình toán học cần phải có cho bộ điều khiển.

Riêng cho trường hợp $y=f(u)$ tuyến tính từng đoạn (gẫy khúc) thì thuật toán tổng hợp sẽ gồm các bước như sau:

1) Xác định các điểm gẫy khúc (u_k, y_k), $k=1, 2, \ldots, n$ của $y=f(u)$.

2) Định nghĩa n tập mờ đầu vào A_k, $k=1, 2, \ldots, n$ có hàm thuộc $\mu_{A_k}(u)$ dạng hình tam giác với đỉnh là điểm u_k và miền xác định là khoảng $[u_{k-1}, u_{k+1}]$, trong đó cho A_1 và A_n thì các điểm u_0, u_{n+1} có thể chọn tùy ý miễn là thỏa mãn $u_0<u_1$ và $u_{n+1}> u_n$.

3) Xác định n tập mờ đầu đầu ra B_k, $k=1, 2, \ldots, n$ có hàm liên thuộc $\mu_{B_k}(u)$ dạng hàm Singleton định nghĩa tại y_k, $k=1, 2, \ldots, n$.

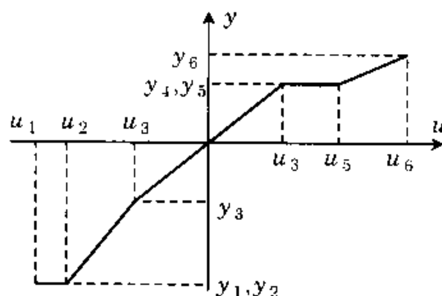
4) Định nghĩa tập n mệnh đề hợp thành

R_i : Nếu $u = A_i$ thì $y = B_i$, $i=1, 2, \ldots, n$.

Như vậy mỗi một giá trị rõ đầu sẽ tích cực được 2 mệnh đề.

5) Sử dụng nguyên tắc độ cao để giải mờ.

Ví dụ 6: Xét ví dụ về đường $y=f(u)$ cho trong hình 17. Đường này có 6 cặp điểm gẫy khúc (u_k, y_k), $k=1, 2, \ldots, 6$.



Hình 17: Đường đặc tính $y=f(u)$ cho trước.

Hình 18 biểu diễn các hàm liên thuộc vào ra của bộ điều khiển mờ có đường đặc tính $y=f(u)$ đã cho trong hình 17.

Luật hợp thành của bộ điều khiển gồm 6 mệnh đề:

R_1 : Nếu $u = A_1$ thì $y = B_1$ hoặc,

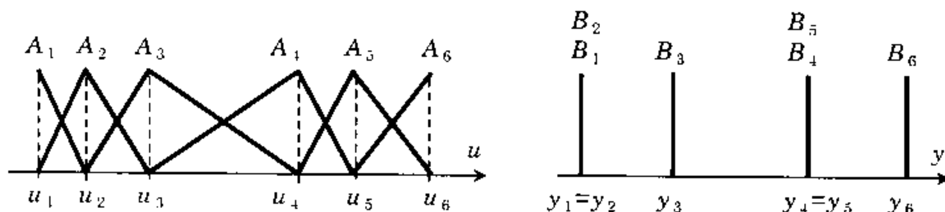
R_2 : Nếu $u = A_2$ thì $y = B_2$ hoặc,

R_3 : Nếu $u = A_3$ thì $y = B_3$ hoặc,

R_4 : Nếu $u = A_4$ thì $y = B_4$ hoặc,

R_5 : Nếu $u = A_5$ thì $y = B_5$ hoặc,

R_6 : Nếu $u = A_6$ thì $y = B_6$.



Hình 18: Hàm liên thuộc của các biến ngôn ngữ vào ra minh họa cho ví dụ 6.

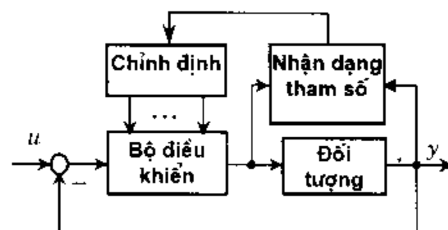
Mở rộng ra, nếu đường $y=f(u)$ không có dạng gẫy khúc, nhưng trơn thì ta có thể xấp xỉ nó bằng một đường gãy khúc $y=\widetilde{f}(u)$ rồi áp dụng thuật toán trên để tìm bộ điều khiển mờ có quan hệ truyền đạt $y=\widetilde{f}(u)$. Do mọi đường trơn $y=f(u)$ đều có thể xấp xỉ bằng một đường gẫy khúc (trong một khoảng kín, giới nội) với độ sai lệch nhỏ tùy ý nên ta có thể khẳng định:

Định lý: Nếu cho trước một hàm trơn $g=g(u)$ trong một miền compact C và một số ε dương nhỏ tuỳ ý thì bao giờ cũng tồn tại một bộ điều khiển mờ có quan hệ truyền đạt $y=f(u)$ thỏa mãn $\sup_{u\in C} |g-y| < \varepsilon$.

Thuật toán trên và như định lý vừa nêu cũng được phát biểu một cách hoàn toàn tương tự cho hàm nhiều biến $y=f(\underline{u})$ để tổng hợp bộ điều khiển mờ MISO khi biết trước quan hệ truyền đạt của nó.

3.3 Cấu trúc bộ điều khiển mờ thông minh (4,10)

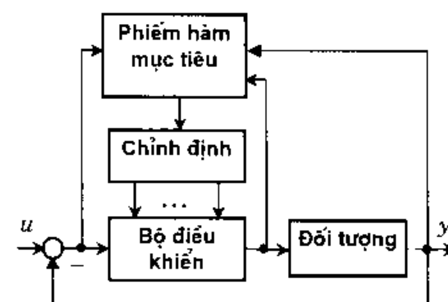
Như đã nói, một trong những tiêu chí hàng đầu thường được quan tâm khi thiết kế bộ điều khiển là tính tự thích nghi với sự thay đổi của đối tượng. Trong thực tế, hệ tự thích nghi được sử dụng nhiều về những ưu điểm của nó so với các hệ thống điều khiển thông thường. Khả năng tự chỉnh định lại các thông số của bộ điều khiển cho phù hợp với đối tượng chưa biết rõ đã đưa hệ tự thích nghi trở thành một *hệ điều khiển thông minh*. So với những bộ điều khiển kinh điển, bộ điều khiển mờ có rất nhiều tham số nên miền chỉnh định cho hệ mờ định hướng thích nghi là rất lớn.



Hình 19: Điều khiển thích nghi trực tiếp

3.3.1 Thích nghi trực tiếp và gián tiếp

Hệ thống điều khiển cơ bản của hệ thích nghi hoàn toàn giống như các hệ thống điều khiển mạch vòng thông thường. Các tính chất của đối tượng dưới tác dụng của điều khiển, thường được tiến hành nhận dạng qua hệ kín hoặc thông qua các đại lượng đặc trưng của hệ như độ quá điều chỉnh cực đại, thời gian quá điều chỉnh cực đại, bình phương sai lệch, tích phân sai số tuyệt đối ....



Hình 20: Điều khiển thích nghi gián tiếp

Mạch vòng thích nghi cho hệ điều khiển mờ hoặc không mờ đều được xây dựng trên 2 phương pháp:

Phương pháp trực tiếp: thực hiện qua

việc nhận dạng thường xuyên các tham số của đối tượng trong hệ kín (hình19). Quá trình nhận dạng thông số của đối tượng có thể thực hiện bằng cách thường xuyên đo trạng thái của tín hiệu vào/ra của đối tượng và chọn một thuật toán nhận dạng hợp lý. Tất nhiên là phải đi kèm với giả thiết là mô hình đối tượng đã biết trước (ví dụ như đối tượng có mô hình của một khâu quán tính bậc một có trễ và các tham số K_p, T_p cần được nhận dạng). Mô hình của đối tượng cũng có thể là mô hình mờ. Mô hình mờ là mô hình biểu diễn dưới dạng câu điều kiện: Nếu ... thì ... hoặc dưới dạng ma trận quan hệ R (ma trận biểu diễn luật hợp thành)

Phương pháp gián tiếp: thực hiện thông qua phiếm hàm mục tiêu của hệ kín xây dựng dựa trên các chỉ tiêu chất lượng. Chất lượng của hệ thống được phản ánh qua các tham số của phiếm hàm mục tiêu. Phiếm hàm mục tiêu có thể được xây dựng dựa trên các chỉ tiêu chất lượng động của hệ thống như độ quá điều chỉnh cực đại, thời gian quá điều chỉnh, các chỉ tiêu ở miền tần số, độ rộng giải thông tần, biên độ cộng hưởng hay các tiêu chuẩn tích phân sai lệch và cũng có thể xây dựng nhiều chỉ tiêu trong cùng một phiếm hàm (hình 20).

3.3.2 Bộ điều khiển mờ tự chỉnh cấu trúc

Các bộ điều khiển mờ thích nghi có khả năng chỉnh định các tham số của tập mờ (các hàm thuộc) gọi là *bộ điều khiển mờ tự chỉnh* (Self-Turning-Controller). Bộ điều khiển mờ có khả năng tự chỉnh định lại các mệnh đề hợp thành (luật điều khiển), ví dụ chuyển từ

Nếu $\mathcal{U}$ = ... thì $\mathcal{Y} = NS$

thành

Nếu $\mathcal{U}$ = ... thì $\mathcal{Y} = ZE$

(sửa đổi phần kết luận) được gọi là *bộ điều khiển mờ tự chỉnh cấu trúc*. Trong trường hợp này, hệ thống có thể bắt đầu làm việc với các luật đã được chỉnh định hoặc với bộ điều khiển còn chưa đủ các luật điều khiển. Các luật điều khiển cần được bổ xung thêm sẽ được thiết lập trong quá trình học.

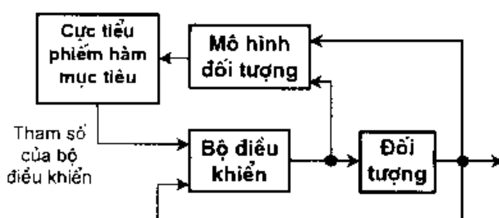
Tóm lại, bộ điều khiển mờ tự chỉnh định các luật điều khiển được gọi là bộ điều khiển mờ tự chỉnh cấu trúc. Bộ chỉnh định được thiết kế đảm bảo đầu ra là giá trị hiệu chỉnh của tín hiệu điều khiển $u(t)$ (tín hiệu ra của bộ điều khiển). Để thay đổi luật điều khiển trước tiên là phải xác định được quan hệ giữa giá trị được hiệu chỉnh ở đầu ra của bộ điều khiển với giá trị biến đổi ở đầu vào. Do vậy cần có mô hình thô của đối tượng, mô hình này dùng để tính toán tương ứng với một giá trị đầu ra cần đạt của bộ điều khiển. Dựa trên tín hiệu ra mong muốn và tín hiệu vào tương ứng của bộ điều khiển có thể xác định và hiệu chỉnh các nguyên tắc điều khiển, các nguyên tắc này đảm bảo chất lượng điều khiển của hệ thống. Một câu hỏi được đặt ra là những giá trị nào của tín hiệu điều khiển $u(t)$ sẽ làm cho chất lượng của hệ thống xấu đi?. Để trả lời được câu hỏi này phải xác định được đặc tính động học của hệ thống. Đối với những đối tượng bậc cao có thời gian trễ lớn có thể có thời gian chỉnh định chậm, còn đối với các hệ thống bậc thấp có thời gian trễ nhỏ yêu cầu thời gian

chỉnh định nhanh. Tóm lại, việc chỉnh định chỉ có ý nghĩa khi quá trình chỉnh định kết thúc trước khi hệ thống kết thúc quá trình quá độ.

3.3.3 Bộ điều khiển mờ tự chỉnh có mô hình theo dõi

Một hệ tự chỉnh không những chỉnh định trực tiếp tham số của bộ điều khiển mà còn chỉnh định cả tham số của mô hình đối tượng được gọi là bộ tự chỉnh có mô hình theo dõi (Model Based Controller MBC). Với bộ điều khiển như vậy hệ mờ không chỉ sử dụng cho quá trình điều khiển đối tượng mà còn phục vụ cho quá trình nhận dạng đối tượng, được gọi là "*mô hình đối tượng mờ*". Hệ tự chỉnh có mô hình theo dõi đã được áp dụng trong hệ thống điều khiển đường tàu điện ngầm ở Sendai/Nhật bản và trong các hệ thống điều khiển mức, các hệ thống mà mức độ khó thực hiện do hằng số thời gian chậm trễ gây ra.

Bộ điều khiển mờ có mô hình theo dõi MBC bao gồm ba thành phần chính:



Hình 21: Điều khiển thích nghi có mô hình theo dõi.

1) Mô hình có đối tượng mờ (thường có dạng quan hệ), được xác định trong khi hệ thống đang làm việc bằng cách đo và phân tích các tín hiệu vào/ra của đối tượng. Vì mô hình của đối tượng gián tiếp xác định các luật hợp thành của bộ điều khiển do vậy bộ điều khiển MBC cũng chính là bộ điều khiển mờ tự chỉnh cấu trúc.
2) Các chỉ tiêu chất lượng được sử dụng trong phiếm hàm mục đích thường được đưa dưới dạng hàm liên thuộc. Thí dụ như trong hệ thống điều khiển mức, độ chênh so với mức mong muốn được biểu diễn bằng hàm liên thuộc dạng hình tam giác, trong đó đỉnh của tam giác chính là giá trị mức mong muốn. Nếu cần tối ưu đồng thời nhiều phiếm hàm mục đích, có thể tổ hợp các chỉ tiêu tương ứng theo toán tử liên kết min.
3) *Lựa chọn tín hiệu điều khiển u* từ tập hợp của các tín hiệu điều khiển xác định từ mô hình đối tượng và đảm bảo chỉ tiêu chất lượng nào đó của hệ thống tốt nhất.

Những bài toán thiết kế theo cấu trúc này thường gặp khi:

- Những thông tin về mô hình đối tượng còn rất ít khi bắt đầu quá trình điều khiển. Bởi vậy thông thường quá trình nhận dạng phải bắt đầu với ma trận quan hệ "rỗng". Theo kinh nghiệm của các phương pháp cũ thì nên bắt đầu với mô hình của đối tượng được nhận dạng ở hệ hở được gọi là *mô hình ban đầu*.

- Trong những trường hợp đặc biệt, ở giai đoạn đầu do thiếu thông tin về đối tượng nên các quyết định điều khiển không thoả mãn được phiếm hàm mục tiêu, hay nói một cách khác là không thỏa mãn được các chỉ tiêu chất lượng đặt ra. Trong những trường hợp như vậy nên thiết kế thêm một bộ điều khiển phụ với chức năng ít nhất là giữ cho hệ thống làm việc ổn định cho đến khi mô hình đối tượng mờ được xác định hoàn toàn. Đơn giản nhất là nên giữ lại giá trị tín hiệu điều khiển $u(t)$ của bước trước đó.

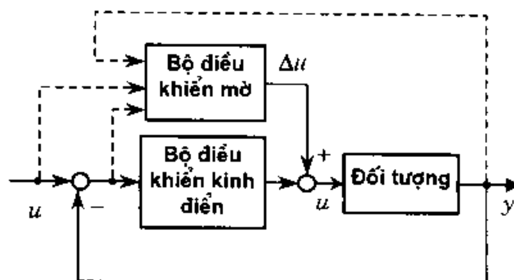
3.3.4 Bộ điều khiển mờ lai

Bộ điều khiển mờ lai (Fuzzy–hybrid) là một bộ điều khiển tự động trong đó thiết bị điều khiển bao gồm hai thành phần:

a) phần thiết bị điều khiển kinh điển.

b) và phần bộ điều khiển mờ.

Phần lớn các hệ thống điều khiển mờ lai là hệ thích nghi, nhưng không phải mọi hệ lai là hệ thích nghi. Ví dụ một hệ thống điều khiển có khâu tiền xử lý để tự chỉnh định tham số bộ điều khiển *một lần* khi bắt đầu khởi tạo hệ thống, sau đó trong suốt quá trình làm việc các thông số đó không được thay đổi nữa, thì không thuộc nhóm các hệ thích nghi. Hoặc một trường hợp khác, hệ thống mà tính "tự thích nghi" của thiết bị điều khiển được thực hiện bằng cách dựa vào sự thay đổi của đối tượng mà chọn khâu điều khiển có tham số thích hợp trong số các khâu cùng cấu trúc nhưng với những tham số khác nhau đã được cài đặt từ trước, cũng không được gọi là hệ điều khiển thích nghi. Tính "*mà vẫn nhầm tưởng là thích nghi*" của các loại hệ thống này được thực hiện bằng cách chuyển công tắc đến bộ điều khiển có tham số phù hợp chứ không phải tự chỉnh định lại tham số của bộ điều khiển đó theo đúng nghĩa của một bộ điều khiển thích nghi đã định nghĩa.



Hình 22: Một ví dụ về cấu trúc hệ mờ lai.

Hình 22 là một ví dụ về hệ mờ lai.

Do bản chất bộ điều khiển mờ chỉ là một bộ điều khiển tĩnh, nên để mang thêm tính động vào cho nó người ta thường phải sử dụng cấu trúc bộ điều khiển mờ lai. Sau đây là một vài bộ điều khiển mờ lai, thích nghi đã được ứng dụng trong công nghiệp [4,9,16,17]:

- Thiết bị kiểm tra các công cụ truyền động.
- Bộ điều khiển cầu treo.
- Bộ điều khiển máy dập khuôn và đóng hộp thuốc viên.
- Bộ phân tích và xử lý tiếng nói.
- Bộ xử lý dữ liệu đo mức bằng sóng cực ngắn.

⋮

Tài liệu tham khảo

[1] **Driankov, D.**: An Introduction to Fuzzy Control. Springer Verlag 1993.

[2] **Dubois, D. and Prade, H.**: Fuzzy Sets and Systems: Theory and Applications. New York: Academic Press, 1980.

[3] **Hellendoorn, H.**: Fuzzy Control: An Overview. Vieweg Verlag, 1994.

[4] **Kahler, J.**: Fuzzy control für Ingenieure. Vieweg Verlag Wiesbaden ,1995.

[5] **Kahlert, J.**: Entwurf, Analyse und Synthese von Fuzzy-Regelungssysteme mit dem Programmsystem WinFACT. 9th Symposium Simulationstechnik ASIM, 1994.

[6] **Klir, G.J. and Yuan, B.**: Fuzzy Sets and Fuzzy Logic: Theory and Applications. Prentice Hall, 1995.

[7] **Kosko, B.**: Neural Networks and Fuzzy Control. Prentice Hall, 1991.

[8] **Kruse, R.; Gebhard, J. and Klawonn, F.**: Foundations of Fuzzy Systems. John Wiley & Sons,1994.

[9] **Lauzi, M.**: Anwendung der Fuzzy-Logic in automatisierungstechnischen Entscheidungsstrukturen. VDI Verlag, 1995.

[10] **Phan Xuân Minh và Nguyễn Doãn Phước:** Lý thuyết điều khiển mờ. (in lần 2). NXB Khoa học và Kỹ thuật, 1999, Hà nội.

[11] Hệ mờ và ứng dụng. Biên tập tập thể : **Nguyễn Hoàng Phương, Bùi Công Cường, Nguyễn Doãn Phước, Phan Xuân Minh và Chu Văn Hỷ.** NXB Khoa học và Kỹ thuật, 1998, Hà nội.

[12] **Sontag, E. D.**: Mathematical Control Theory. Spring Verlag New York, 1990.

[13] **Wang, L.X.**: A Course in Fuzzy Systems and Control. Prentice-Hall International, Inc. 1997.

[14] **Wechler, W.**: The Concept of Fuzziness in Automata and Language Theory. Akademie-Verlag, 1978.

[15] **Zimmermann, H.J.**: Fuzzy-Set-Theory and its Applications. Boston, Dordrecht, London: Kluwer Academic Publishers, 1991.

[16] **Zimmermann, H.J. , Altrock, C.**: Fuzzy Logic Anwendungen. Oldenburg Verlag, München, 1995.

[17] **Zadeh,L., Jamshidi M., Titli A., Boverie S.**: Application of Fuzzy Logic. Prentice Hall PTR, New Jersey. 1997.

5 ĐIỀU KHIỂN ƯỚC LƯỢNG VÀ MÔ HÌNH TRÊN CƠ SỞ ĐIỀU KHIỂN MỜ

Phan Xuân Minh & Nguyễn Doãn Phước
Đại học Bách khoa Hà Nội

Hai thập kỷ cuối của thế kỷ XX, với sự phát triển mạnh mẽ của công nghệ thông tin nhất là kỹ thuật vi xử lý và công nghệ phần mềm đã đặt nền móng cho việc ứng dụng hệ thống điều khiển thông minh vào các ngành công nghiệp. Các hệ thống điều khiển thông minh được xây dựng trên cơ sở trí tuệ nhân tạo đã giúp cho con người có khả năng chế ngự được những đối tượng mà trước kia tưởng chừng như không điều khiển được. Một trong những hệ thống điều khiển thông minh đó là hệ thống điều khiển mờ, hệ thống điều khiển được thiết kế dựa trên cơ sở toán học là logic mờ do Zadeh đặt nền móng.

Điều khiển mờ được chia thành hai lớp bài toán điều khiển đó là điều khiển ước lượng mờ và mô hình mờ. Điều khiển ước lượng mờ được áp dụng cho các bài toán điều khiển mà đối tượng điều khiển có mô hình không chính xác hoặc không tường minh hay nói một cách khác là lượng thông tin về đối tượng không đầy đủ, trong khi mô hình mờ là bài toán xây dựng mô hình cho đối tượng theo phương pháp mờ.

Những thuật toán điều khiển mờ đang được quan tâm và đã đạt được những kết quả khả quan trong nhiều ứng dụng công nghiệp đó là:

- Điều khiển Mamdani (MC-Mamdani Control)
- Điều khiển mờ trượt (SMFC-Sliding Mode Fuzzy Control)
- Điều khiển tra bảng (CM-Cell Mapping Control)
- Điều khiển Takagi/Sugeno (TS-Control)
- Điều khiển Takagi/Sugano với phương pháp tuyến tính hoá của Lyapunov

1 Điều khiển Mamdani

Cách thức làm việc của các bộ điều khiển mờ FC theo kiểu này hoàn toàn dựa trên kinh nghiệm của các chuyên gia. Vì không thể xây dựng được mô hình của đối tượng một cách chính xác nên không thể thiết kế bộ điều khiển theo các phương pháp thông thường. Tuy nhiên vẫn có thể thiết kế theo nguyên tắc sai lệch bằng cách cài đặt có chọn lọc các luật điều khiển cho bộ điều khiển mờ FC. Như vậy mọi hoạt động của đối tượng được điều khiển thông qua các luật điều khiển (Control Rule). Nhìn nhận theo quan điểm điều khiển, một hệ thống như vậy không thể thoả mãn được các chỉ tiêu chất lượng đặt ra cho hệ thống. Để khắc phục nhược điểm trên, trong quá trình điều khiển có thể kết hợp xây dựng mô hình toán học gần đúng cho đối tượng theo luật mờ.

Điều khiển ước lượng còn gọi là cơ chế suy diễn Mamdani được sử dụng trong trường hợp cả mệnh đề nguyên nhân và mệnh đề kết quả đều là các giá trị mờ. Một luật điều khiển như vậy được biểu diễn cho một hệ SISO như sau:

R_i: Nếu $x = LX^i$ thì $u = LU^i$

và cho một hệ MISO bao gồm hai biến vào và một biến ra có dạng sau:

R_i: Nếu $x = PS$ và $\dot{x} = NB$ thì $u = PM$

Luật điều khiển này cũng có thể biểu diễn dưới dạng vector các biến vào:

R_i: Nếu $\underline{x} = \begin{pmatrix} x \\ \dot{x} \end{pmatrix} = \begin{pmatrix} PS \\ NB \end{pmatrix}$ thì $u = PM$

trong đó $\underline{x} = \begin{pmatrix} x \\ \dot{x} \end{pmatrix}$ là vector biến đầu vào có giá trị mờ $LX^i = \begin{pmatrix} PS \\ NB \end{pmatrix}$ và biến đầu ra $u \in U$ có giá trị mờ là $LU^i = PM$.

Thậm chí ngay cả khi hiểu biết về hệ thống còn rất ít vẫn có thể thiết kế được luật điều khiển u nếu như có được một vài ý tưởng về trạng thái động của hệ qua trạng thái của x và đạo hàm của nó $\dot{x}$.

Thông thường các hiểu biết thường được thể hiện dưới dạng cấu trúc nên hoàn toàn có thể biểu diễn bằng các luật mờ. Một luật mờ đặc trưng trong việc xây dựng mô hình là nếu như biết được x và đại lượng điều khiển u thì cũng xác định được giá trị của đạo hàm của x theo:

R_{si}: Nếu $\underline{x} = LX^i$ và $u = LU^i$ thì $\underline{\dot{x}} = L\dot{X}^i$

hay

R_{si}: Nếu $(x, \dot{x})^T = (PS, NB)^T$ và $u = PM$ thì $(\dot{x}, \ddot{x}) = (NM, PM)^T$

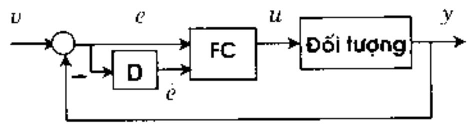
với $\underline{x} = (x, \dot{x})^T$, $LX^i = (PS, NB)^T$, $\underline{\dot{x}} = (\dot{x}, \ddot{x}) = (NM, PM)^T$, $u = LU^i = PM \in U$.

Một bộ điều khiển mờ trượt (Sliding Mode Fuzzy Control-SMFC) cũng là một bộ điều khiển Mamdani. Mặt khác nếu hiểu biết tương đối về đối tượng, về cấu trúc của hệ thống thì vẫn có thể có được những kinh nghiệm tương ứng để điều khiển hệ thống. Hiểu biết tương đối về đối tượng được thể hiện trong thiết kế là có khả năng xây dựng các luật điều khiển đảm bảo cho hệ thống làm việc một cách bình thường. Hay nói một cách khác, là tạo ra được bảng vào/ra của bộ điều khiển và của hệ thống đầy đủ và chính xác đảm bảo hoạt động bình thường của hệ thống. Đối với hệ thống đó chính là bài toán nhận dạng tham số của hệ thống.

2 Điều khiển mờ trượt (Sliding Mode FC)

Phần lớn các đối tượng phi tuyến bậc hai được thiết kế dựa trên phương pháp mặt phẳng pha. Các quyết định điều khiển được đưa ra dựa trên phân tích tín hiệu sai lệch $e(t)$ và đạo hàm của nó theo thời gian $\dot{e}(t)$.

Tín hiệu điều khiển u từ FC được xác định từ các giá trị của sai lệch $e(t)$ và đạo hàm của nó $\dot{e}(t)$ qua các luật điều khiển mờ. Quan điểm mô phỏng để thiết kế các luật điều khiển mờ này hoàn toàn dựa trên đường chuyển đổi (trượt) của hệ trong mặt phẳng pha. Điều đó có nghĩa là bộ điều khiển mờ điều khiển theo kiểu đường chéo (diagonal form). Một khả năng thiết kế để đảm bảo hệ trượt trơn về điểm cân bằng là đường chuyển đổi đồng thời là đường quỹ đạo pha của hệ thống (time optimale trajectory).



Hình 1: Hệ mờ trượt.

Luật điều khiển mờ theo dạng đường chéo thường được biểu diễn như sau:

	NB	*NM*	*NS*	*Z*	*PS*	*PM*	*PB*
PB	*Z*	*NS*	*NS*	*NM*	*NM*	*NB*	*NB*
PM	*PS*	*Z*	*NS*	*NS*	*NM*	*NM*	*NB*
PS	*PS*	*PS*	*Z*	*NS*	*NS*	*NM*	*NM*
Z	*PM*	*PS*	*PS*	*Z*	*NS*	*NS*	*NM*
NS	*PM*	*PM*	*PS*	*PS*	*Z*	*NS*	*NS*
NM	*PB*	*PM*	*PM*	*PS*	*PS*	*Z*	*NS*
NB	*PB*	*PB*	*PM*	*PM*	*PS*	*PS*	*Z*

P–positiv
N–negativ
Z–zero
S–small
M–medium
B–big

Với luật hợp thành kiểu:

R : Nếu $e=PS$ và $\dot{e}=NB$ thì $u=PS$

trong đó, chẳng hạn như *PS* là giá trị mờ *dương ít* (POSITIV SMALL) của biến sai lệch e, *NB* là giá trị mờ *âm ít* (NEGATIV SMALL) của biến $\dot{e}$ và *PM* là giá trị mờ *dương vừa* (POSITIV MEDIUM) của biến điều khiển u. Ở điều khiển kinh điển, u chỉ có một trong hai giá trị hoặc dương hoặc âm được quyết định thông qua các giá trị mờ của e và $\dot{e}$, thực chất là qua dấu của $s=\lambda e+\dot{e}$. Nhưng trong trường hợp này tín hiệu điều khiển u được tạo ra có nhiều giá trị mờ (*NB*, *NM*, *NS*, *Z*, *PS*, *PM*, *PB*) phụ thuộc vào dấu và khoảng cách $|s|$ giữa trạng thái thực tế của e với đường chuyển đổi $s=\lambda e+\dot{e}=0$.

Dễ dàng nhận thấy rằng, phương pháp thiết kế này rất gần với phương pháp thiết kế hệ trượt có giới hạn (SMC with a boundary layer BL), đó là một phương pháp thiết kế bộ điều khiển bền vững. Điều khiển trượt được ứng dụng trong thiết kế điều khiển các hệ phi tuyến có mô hình không chính xác (uncertainties), tham số không xác định và làm việc trong môi trường có nhiễu tác động. Sự giống nhau giữa điều khiển mờ theo đường chéo (diagonal FC) và điều khiển trượt có giới hạn(SMC with BL) cho phép người thiết kế tạo ra bộ điều khiển mờ theo đường chéo hoạt động hoàn toàn giống như bộ điều khiển trượt có giới hạn, và vì vậy bộ điều khiển mờ hoàn toàn đảm bảo được tính ổn định, bền vững như bộ điều khiển trượt có giới hạn. Bộ điều khiển mờ thiết kế như trên được gọi là SMFC.

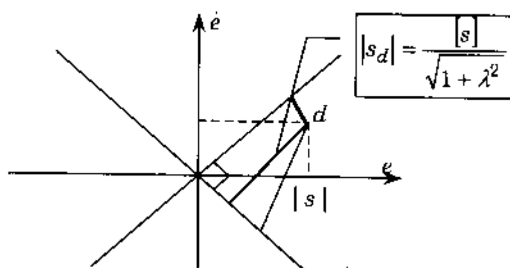
Tuy nhiên, cũng có câu hỏi đặt ra rằng: đường chuyển đổi của bộ SMFC có dạng như thế nào? Câu trả lời hoàn toàn đơn giản vì thực chất bộ điều khiển trượt có giới hạn là một trường hợp đặc biệt của SMFC. Điều khiển trượt có giới hạn có đặc tính truyền đạt tĩnh tuyến tính và bị giới hạn về hai phía trên và dưới, còn SMFC không nhất thiết phải đường thẳng giữa hai giới hạn hoàn toàn có thể là đường cong và có thể thay đổi cho phù hợp với đối tượng. Ví dụ như tạo ra một đặc tính động cho hệ thống kín với thời gian quá độ ngắn và độ quá điều chỉnh nhỏ hoàn toàn là những yêu cầu có khả năng thực hiện được khi sử dụng SMFC. Điều đó có thể đạt được khi hiệu chỉnh hệ số khuếch đại của bộ điều khiển theo nguyên tắc là càng xa đường chuyển đổi thì hệ số khuếch đại càng lớn.

Do cách thiết kế theo nguyên tắc đường chéo nên các luật mờ có thể định nghĩa lại dựa trên khoảng cách $|s|$ giữa vectơ trạng thái sai lệch e và đường trượt $s = \lambda e + \dot{e} = 0$ và giá trị mờ đầu ra u của SMFC được chọn phù hợp với khoảng cách này. Bằng cách thiết kế như vậy sẽ giảm được đáng kể số lượng các luật điều khiển mờ R_i đặc biệt trong các trường hợp bậc của hệ thống cao. Ví dụ trong trường hợp hệ bậc 2, mỗi biến trạng thái gồm m tập mờ, nếu theo phương pháp đường chéo ta phải thiết kế m^2 luật suy diễn mờ R_i cho bộ FC, nhưng nếu thiết kế theo phương pháp SMFC thì chỉ cần thiết kế m luật R_i là đủ. Tại sao lại có thể thiết kế như vậy như vậy? -Điều đơn giản là vì SMFC chỉ mô tả lại quan hệ giữa khoảng cách của véc tơ trạng thái đến đương trượt và dựa trên quan hệ đó ra tạo ra một giá trị điều khiển u phù hợp.

Ngoài ra cũng có thể xây dựng các luật điều khiển mờ của SMFC dựa trên quan hệ giữa véc tơ trạng thái với đường thẳng trực giao của đường trượt đi qua gốc toạ độ. Luật điều khiển mờ trong trường hợp này được biểu diễn dưới dạng:

R : Nếu $e=PS$ và $d=S$ thì $u=NS$

Ưu điểm nổi bật của SMFC là ở chỗ không những tìm được giải pháp thực hiện điều khiển đối với số lượng lớn các bài toán điều khiển mà hệ thống có bộ điều khiển được thiết kế như vậy còn đảm bảo tính bền vững. Để làm giảm sai lệch tĩnh của toàn hệ ở chế độ xác lập, ta tích hợp thêm vào SMFC một bộ tích phân cho các đối tượng có nhiều nhiễu tác động và có mô hình không chính xác (uncertainly). Có nhiều cách khác nhau để thực hiện ý tưởng này. Ta có thể tích hợp thêm một biến ngôn ngữ đại diện cho thành phần tích phân vào SMFC hoặc có thể thay đổi chế độ trượt như ở trên đã trình bày.



Hình 2: Xác định khoảng cách.

Còn một vấn đề cần giải quyết nữa là số hoá các biến vào của SMFC trước khi mờ hoá. Phương pháp số hoá quyết định chất lượng của bộ điều khiển, nó đòi hỏi phải phản ảnh trung thành đại lượng vật lý quan sát được từ đối tượng và đưa ra giải pháp chọn các giá trị

mờ tối ưu đối với các biến mờ đầu vào. Giải pháp thông thường là chuẩn hoá tính hiệu bằng một thừa số k_c và tín hiệu điều khiển u_c chuẩn hoá được xác định từ SMFC có đầu vào chuẩn hoá. Tín hiệu ra u chính bằng:

$$u = k_g u_c \ .$$

Chọn thừa số chuẩn hoá k_c không chỉ thuần tuý giải quyết vấn đề chuẩn hoá các biến vào mà còn quan trọng trong điều khiển thích nghi và chỉnh định trực tuyến của SMFC. Đặc tính động học của hệ kín hoàn toàn phụ thuộc vào đặc tính tĩnh (mặt phẳng trượt) của SMFC. Mặt trượt này được xác định thông qua các hàm liên thuộc của biến mờ vào/ra chuẩn hoá được định nghĩa bằng các giá trị mờ của SMFC. Do vậy khi chọn thừa số k_c và k_g cần phải lưu ý các điểm sau đây:

1) Chọn k_g cho tín hiệu điều khiển u phải quan tâm nhất đến tính ổn định và dao động của hệ, bởi vì trong thiết kế tính ổn định của hệ thống có mức độ ưu tiên cao nhất.
2) Chọn k_c của SMFC cần quan tâm đến độ nhạy được phản ánh qua vùng làm việc của hai tham số s và d của hệ thống. k_c có mức độ ưu tiên thứ hai trong thiết kế.
3) Chọn hàm liên thuộc và qua đó quan hệ vào/ ra tĩnh của SMFC được xác định. Quan hệ này quyết định trạng thái mờ của hệ ở miền hoạt động của đường trượt s và tín hiệu điều khiển mờ u và có thể chỉnh định được thông qua hệ số chuẩn hoá k_c và k_g . Chỉ tiêu chất lượng này có mức độ ưu tiên thứ ba trong thiết kế.

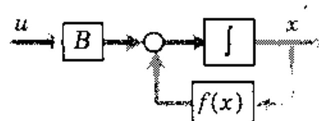
Vấn đề cuối cùng là thiết kế SMFC cho hệ MIMO. Hệ mờ SISO được coi là một trường hợp đặc biệt của hệ MIMO. Một hệ mờ MIMO là hệ có nhiều tín hiệu điều khiển u_i và nhiều biến trạng thái đầu vào x_i . Đây là một hệ mà tham số của mô hình không xác định, hoàn toàn khác với hệ thống không tham số (không có mô hình).

Cho đối tượng có mô hình phi tuyến:

$$\dot{x} = f(x) + Bu \ ,$$

$f(x)$ là véc tơ hàm phi tuyến của véc tơ trạng thái x, u là véc tơ tín hiệu đầu vào và B là ma trận đầu vào.

Điều kiện cần thiết để thiết kế được SMFC đảm bảo tính ổn định trong trường hợp tham số không xác định là ma trận B là ma trận vuông đảm bảo tính điều khiển được của hệ và tồn tại ma trận nghịch đảo B^{-1}.



Hình 3: Đối tượng phi tuyến.

3 Điều khiển tra bảng (Cell mapping)

Người đặt nền móng cho phương pháp thiết kế điều khiển này là C.S.Hsu. Điều khiển theo bảng được bắt nguồn từ kỹ thuật tính với những kết quả ban đầu là tính toán trạng thái và khảo sát tính ổn định của các hệ thống phi tuyến. Những kết quả này đã khẳng định khả năng xây dựng các mô hình bằng phương pháp ước lượng trên máy tính và ngày nay đã trở

thành một định hướng nghiên cứu được các nhà khoa học và công nghệ đặc biệt quan tâm. Những người đầu tiên ứng dụng phương pháp điều khiển tra bảng vào hệ mờ là Chen và Tsao. Sau đây là những ưu điểm chính của điều khiển tra bảng:

- Cung cấp các chiến lược tự học cho FC
- Các phương pháp thiết kế FC tối ưu về mặt thời gian.

Ý tưởng của Hsu bắt đầu bằng việc biểu diễn rời rạc hệ phi tuyến theo phương trình:

$$x(k+1)=f(x(t_k),u(t_k)), \tag{4.1}$$

trong đó t_k là điểm thời gian rời rạc mà tại đó quá trình tra bảng được thực hiện. Thời gian t_k không nhất thiết phải trùng với thời gian của hệ. Thông thường để tra bảng người ta chia không gian trạng thái thành số lượng xác định các phần tử (cell). Mỗi phần tử biểu diễn trục x_i của không gian trạng thái trong miền giới hạn s_i, được thể hiện thông qua giá trị z_i. Nếu chia không gian trạng thái thành a phần tử, thì a phần tử là n−tuple (véc tơ) của $z=(z_1,z_2,...,z_n)^T$. Phần không gian trạng thái không được quan tâm (nằm ngoài bảng) được gọi là phần bỏ đi (sink cell). Trạng thái của hệ (4.1) được mô phỏng qua z với điểm trọng tâm x^C. Bảng tra C được định nghĩa như sau:

$$z(t_{k+1})= C((z(t_k)). \tag{4.2}$$

Từ phương trình trạng thái rời rạc của hệ (4.1) điểm biểu diễn ảo của điểm trạng thái $x(t_k)$ được tính toán và được xác định bằng một phần tử trong bảng đã được xếp đặt sẵn. Điều hiển nhiên là không phải tất cả các điểm thuộc $x(t_k)$ trong $z(t_k)$ đều có cùng phần tử ảo $z(t_{k+1})$. Tuy nhiên phần tử ảo của điểm trạng thái trọng tâm x^C thì chắc chắn tồn tại. Một phần tử (cell) sau khi tra bảng lại chính là nó được gọi là phần tử cân bằng (equilibrium cell). Tất cả các phần tử trong không gian trạng thái hữu hạn được gọi là regular cell.

Điểm mạnh của điều khiển tra bảng là tạo ra tín hiệu điều khiển $u(t_k)$ lái hệ thống đến điểm trạng thái mong muốn đồng thời thoả mãn chỉ tiêu chất lượng tối ưu đặt ra. Mỗi một phần tử trong bảng (một cell) có những tính chất sau đây:

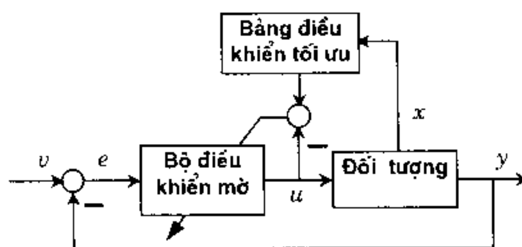
- Số nhóm $G(z)$ quyết định số phần tử z phụ thuộc vào miền dự đoán hoặc miền hấp dẫn.
- Số bước $S(z)$ quyết định số lần chuyển đổi cần thiết để chuyển từ phần tử z sang phần tử dự đoán (preodic cell).
- Số lượng dự đoán $P(z)$ quyết định số phần tử cấu thành chuyển động dự đoán.

Tính chất này được giới thiệu trong thuật toán tìm kiếm chuyển động dự đoán và miền hấp dẫn.

Ứng dụng của phương pháp này trong điều khiển mờ thông qua các phần tử (cell) mà trong đó mỗi phần tử mô tả trạng thái của hệ thống phụ thuộc vào các luật của hệ mờ tương

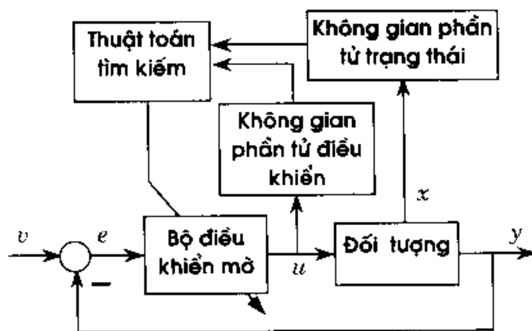
ứng. Thêm nữa mỗi một phần tử còn mô tả một tác động điều khiển phụ thuộc vào luật điều khiển mờ tương ứng.

Smith và Conner phát triển thuật toán điều khiển tra bảng mờ, tư tưởng chính của phương pháp là chỉnh định bộ điều khiển mờ trên cơ sở bảng trạng thái của hệ. Mỗi một phần tử trong bảng tương ứng với một tác động điều khiển đồng thời là điều khiển tối ưu của một phiếm hàm mục đích nào đó, ví dụ như là nghiệm của bài toán tối ưu tác động nhanh. Với một phiếm hàm mục đích và một mô hình của đối tượng, thuật toán phần tử trạng thái (cell state space), phát ra một bảng mô tả các tác động điều khiển trên cơ sở quĩ đạo điều khiển tối ưu. Bộ điều khiển mờ FC thực hiện tra từ phần tử trạng thái ra phần tử tác động điều khiển. Kỹ thuật tra bảng từ phần tử này ra phần tử kia đã được Takagi và Sugano sử dụng trong việc chỉnh định bộ điều khiển của mình (hình 4).



Hình 4: Điều khiển tra bảng theo Smith và Comer

Kang và Vachtsevanos đã phát triển thuật toán gán chân dung pha trên quan hệ giữa phần tử trạng thái và phần tử điều khiển (hình 5). Trên quan điểm đó, họ đã tách riêng biệt không gian của các phần tử trạng thái và không gian của phần tử điều khiển. Không gian của phần tử trạng thái x-cell được ghi lại trong quá trình mô phỏng với tín hiệu vào là hằng. Thuật toán tìm kiếm đảm bảo chỉnh định các luật của bộ điều khiển mờ sao cho hệ ổn định tiệm cận. Nó giải quyết vấn đề xác định điều khiển tối ưu không phụ thuộc vào điểm tìm kiếm đầu tiên.



Hình 5 : Điều khiển tra bảng theo Kang và Vachtsvanos

Hu, Tai và Shenoi sử dụng thuật giải di truyền để thực hiện thuật toán tra bảng. Mục đích của phương pháp là để chỉnh định bộ điều khiển của Takagi và Sugano.

4 Mô hình TS trên cơ sở điều khiển mờ

Thiết kế bộ điều khiển mờ trên cơ sở mô hình bắt đầu từ những hiểu biết trên cơ sở toán học về hệ thống để điều khiển hệ. Sẽ có câu hỏi được đặt ra: tại sao lại thiết kế điều khiển mờ trong khi các bộ điều khiển kinh điển hoàn toàn có thể thực hiện nhiệm vụ điều khiển này. Những nguyên nhân dẫn đến việc thiết kế bộ điều khiển mờ mặc dù đã có mô hình toán học của đối tượng là bởi vì:

1) Điều khiển mờ là một phương pháp điều khiển đơn giảnvà thân thiện với người thiết kế.
2) Điều khiển mờ cung cấp những chiến lược điều khiển phi tuyến, những chiến lược này có quan hệ mật thiết với kỹ thuật điều khiển hệ phi tuyến truyền thống.
3) Đặc tính phi tuyến tĩnh của bộ điều khiển mờ có thể thay đổi linh hoạt khi thay đổi vị trí và dạng của hàm liên thuộc do vậy có thể dễ dàng áp dụng các thuật toán thích nghi.
4) Khả năng xấp xỉ của bộ điều khiển mờ cho phép thiết kế luật điều khiển cho toàn bộ hệ dựa trên cơ sở trợ giúp chỉ của vài luật cơ bản.
5) Có thể thiết kế bộ điều khiển mờ trên cơ sở kỹ thuật tra bảng tỷ lệ (gain sheduling technique). Trong trường hợp này bộ điều khiển mờ được sử dụng như một bộ xấp xỉ luật giữa các luật điều khiển tuyến tính.

Mô tả hệ bắt đầu từ mô hình mờ sử dụng cả không gian trạng thái mờ và mô tả linh hoạt của hệ. Nguyên tắc của hệ Takagi/Sugano (TS) có thể được diễn giải qua ví dụ dưới đây:

Một hệ TS bao gồm hai luật điều khiển với hai đầu vào x_1. x_1 và một đầu ra y.

R_1: Nếu $x_1 = BIG$ và $x_2 = MEDIUM$ thì $y_1 = x_1 - 3x_2$

R_2: Nếu $x_1 = SMALL$ và $x_2 = BIG$ thì $y_2 = 4 + 2x_1$

Đầu vào rõ đo được là: $x_1^* = 4$ và $x_2^* = 60$. Từ hình 3 ta xác định được:

$$LX_{BIG}(x_1^*) = 0,3 \quad \text{và} \quad LX_{BIG}(x_2^*) = 0,35$$

$$LX_{SMALL}(x_1^*) = 0,7 \quad \text{và} \quad LX_{MEDIUM}(x_2^*) = 0,75$$

Từ đó ta xác định được :

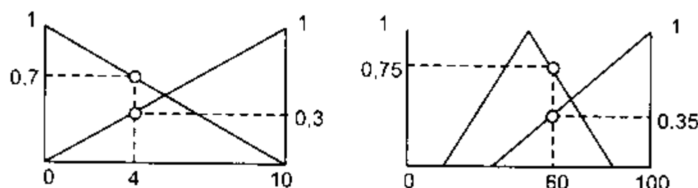
$$\min(0,3\ ;\ 0,75) = 0,3 \quad \text{và} \quad \min(0,7\ ;\ 0,35) = 0,35$$

và

$$y_1 = 4 - 3 \cdot 60 = -176, \qquad y_2 = 4 + 2 \cdot 4 = 12.$$

Như vậy hai thành phần luật R_1 và R_2 là $(0,3\ ;\ -176)$ và $(0,35\ ;\ 12)$. Theo phương pháp tổng trọng số trung bình (weighted normalized sum) ta có:

$$y = \frac{0,3.(-174) + 0,35.12}{0,3 + 0,35} = -74,77$$



Hình 6: Minh họa cho ví dụ

Phương pháp này có thể mở rộng để biểu diễn hệ dưới dạng phương trình vi phân, ví dụ như một vùng mờ LX^i được mô tả bởi luật:

$$R_{si}: \quad \text{Nếu } x = LX^i \text{ thì } \dot{x} = A(x^i)x + B(x^i)u \tag{5.1}$$

Luật này có nghĩa là: nếu véc tơ trạng thái x nằm trong vùng LX^i thì hệ thống được mô tả bởi phương trình vi phân cục bộ $\dot{x} = A(x^i)x + B(x^i)u$. Nếu toàn bộ các luật hệ thống được xây dựng thì có thể mô tả trạng thái của hệ trong toàn cục. Trong (5.1) ma trận $A(x^i)$ và $B(x^i)$ là những ma trận hằng của hệ thống ở trọng tâm của miền LX^i được xác định từ các chương trình nhận dạng.

Kết quả thu được là:

$$\dot{x} = \sum w_i \left(A(x^i)x + B(x^i)u \right) \tag{5.2}$$

với $w_i(x) \in [0,1]$ là độ thoả mãn đã chuẩn hoá của x^* đối với vùng mờ LX^i.

Luật điều khiển tương ứng sẽ là:

$$R_{ci}: \quad \text{Nếu } x = LX^i \text{ thì } u = K(x^i)x \tag{5.3}$$

và luật điều khiển cho toàn bộ không gian trạng thái có dạng:

$$u = \sum_{i=1}^{n} w_i(x) K(x^i) x \,. \tag{5.4}$$

Kết hợp (5.2) và (5.4) ta có phương trình động học cho hệ kín:

$$\dot{x} = \sum w_i(x) w_j(x) \left(A(x^i) + B(x^i) K(x^j) \right) x \,. \tag{5.5}$$

Cần phải nhận mạnh rằng hệ thống được mô tả bằng tập các luật như (5.2) là một hệ phi tuyến thậm chí ngay cả khi ở lân cận trọng tâm của vùng. Trong thực tế $w_i(x)$ phụ thuộc vào véc tơ trạng thái x. Thậm chí ngay cả khi $w_i(x)$ là một hàm tuyến tính hoá từng đoạn đối với véc tơ trạng thái x, thì tích $w_i(x) w_j(x) \left(A(x^i) + B(x^i) K(x^j) \right) x$ ở (5.5) vẫn luôn là hàm phi tuyến.

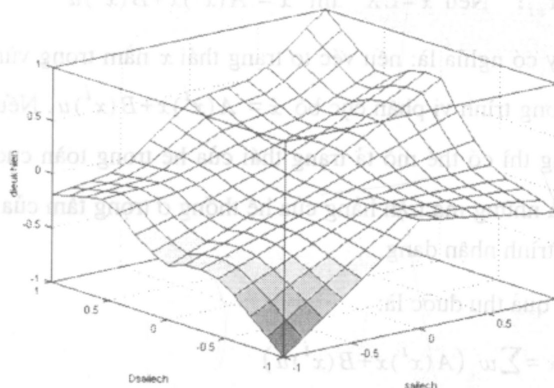
6 Mô hình trên cơ sở điều khiển mờ với phương pháp tuyến tính hoá của Lyapunov

Phần này trình bày về hệ thống điều khiển bao gồm đối tượng điều khiển có mô hình toán học biết trước và bộ điều khiển mờ FC. Trong trường hợp này đối tượng và bộ điều khiển được mô tả toán học ở những mức độ khác nhau. Đối tượng có mô hình khởi đầu là:

$$\dot{x} = f(x, u) \tag{6.1}$$

và bộ điều khiển với các luật mờ:

$$R_{ci}: \quad \text{Nếu } x = LX^i \text{ thì } u = LU^i \tag{6.2}$$



Hình 7 : Quan hệ truyền đạt.

Để có thể nghiên cứu về tính ổn định, tính bền vững và giải quyết các vấn đề với hệ kín, người ta phải mô tả toán học đối tượng và bộ điều khiển ở cùng một mức độ. Chính vì vậy, người ta chuyển đổi cách biểu diễn bộ điều khiển mờ bằng mô hình có cấu trúc:

$$u = g(x), \tag{6.3}$$

với $g(x)$ là quan hệ truyền đạt (tĩnh) giữa vector trạng thái x và vector điều khiển u. Quan hệ truyền đạt cung cấp thông tin toàn bộ và cục bộ về bộ điều khiển. Từ những thông tin này người ta có thể kết luận được hệ thống ổn định hay không ổn định. Ngoài ra người ta còn cảm nhận được quĩ đạo trạng thái của hệ trong không gian trạng thái.

Mặt khác, để nghiên cứu trạng thái cục bộ của hệ ở lân cận một điểm đặc biệt nào đó người ta tuyến tính hoá mô hình toán học của hệ tại lân cận điểm đặc biệt. Ví dụ như hệ (6.1) được tuyến tính hoá xung quanh điểm trạng thái x^d và véc tơ điều khiển u^d, ta có

$$\dot{x} = f(x^d, u^d) + A(x^d, u^d)(x - x^d) + B(x^d, u^d)(u - u^d) \tag{6.4}$$

với $A(x^d, u^d) = \dfrac{\partial f(x,u)}{\partial x}$ và $B(x^d, u^d) = \dfrac{\partial f(x,u)}{\partial u}$ tính tại $x = x^d$ và $u = u^d$ là những ma trận Jacobi.

Luật điều khiển u được xấp xỉ bằng công thức:

$$u = u^d + K(x^d)(x - x^d), \tag{6.5}$$

với $K(x^d)$ là ma trận hệ số khuếch đại tỷ lệ. Khi hệ (6.1) thay đổi trạng thái với điểm đặt trước x^d, thì luật điều khiển (6.5) cũng thay đổi đồng thời với sự thay đổi của x^d. Để thiết kế luật điều khiển cho hệ kín ứng với mỗi điểm x^d bất kỳ người ta xấp xỉ (6.4) bằng tập của các luật mờ TS

$$R_{ci}: \quad \text{Nếu } x^d = LX^i \text{ thì } \dot{x} = f(x^d, u^d) + A(x^d, u^d)(x - x^d) + B(x^d, u^d)(u - u^d)$$

Phương trình động học của hệ thống có dạng:

$$\dot{x}=f(x^d,u^d)+\sum_{i=1}^{n}w_i(x^d)A(x^d,u^d)(x-x^d)+B(x^d,u^d)(u-u^d) \tag{6.6}$$

là một phương trình vi phân tuyến tính vì trọng số w_i chỉ phụ thuộc vào véc tơ trạng thái x^d mà không phụ thuộc vào véc tơ trạng thái x.

Tập luật điều khiển tương ứng là:

R_{ci}: Nếu $x^d=LX^i$ thì $u=u^d+K(x^i)(x-x^d)$ (6.7)

và luật điều khiển là:

$$u=u^d+\sum_{i=1}^{n}w_i(x^d)K(x^i)(x-x^d) \tag{6.8}$$

Thay (6.8) vào (6.6), ta được phương trình động học của hệ kín:

$$\dot{x}=f(x^d,u^d)+\sum_{i,j=1}^{n}w_i(x^d)w_j(x^d)\big(A(x^d,u^d)+B(x^d,u^d)K(x^j)\big)(x-x^d). \tag{6.9}$$

Biểu diễn $A(x^i,u^i)+B(x^i,u^i)K(x^j)$ bằng ma trận A_{ij}, điều kiện đảm bảo ổn định tiệm cận cho hệ (6.9) là tồn tại một ma trận P xác định dương luôn luôn thoả mãn chỉ tiêu chất lượng nội của Lyapunov

$$A_{ij}^T P+PA_{ij}<0$$

với A_{ij} là ma trận Hurwitz. Với kếtquả này, người ta có thể nghiên cứu tính chất ổn định, bền vững và giải quyết nhiệm vụ điều khiển đối với hệ kín lân cận điểm đặt trước x^d tuỳ chọn so với điểm trạng thái được định nghĩa trước đó x^i.

Tài liệu tham khảo

[1] **Phan Xuân Minh & Nguyễn Doãn Phước:** Lý thuyết điều khiển mờ, NXB KH & KT 2000.

[2] **Hung T. Nguyen, Michio Sugano:** Fuzzy Systems. Modeling and Control, Kluwer academic publishers 1998.

[3] **Christopher Edwards and Sarah K. Spurgeon, Taylor and Fransis:** Sliding Mode Control Theory and Application, 1998.

[4] Control Engineering, Addison-Wesley 1990

[5] **C. J. Harris, C. G. Moore & M. Brown:** Intelligent control. Aspects of Fuzzy Logic and Neural Nets. World Sciencific 1993

[6] **Chin-Teng Lin and C.S. George Lee:** Neural Fuzzy Systems, Prentice-Hall International 1996

[7] **J. Kahler:** Fuzzy Control fuer Ingeneure, Vieweg Verlag 1994

[8] **L. X. Wang:** A Course in Fuzzy System and Control, Prentice-Hall International 1997

6 NHẬP MÔN MẠNG NƠ-RON

Nguyễn Doãn Phước & Phan Xuân Minh
Đại học Bách khoa Hà Nội

Với logic mờ, trí tuệ nhân tạo phát triển mạnh mẽ trong những năm gần đây tạo ra cơ sở xây dựng các hệ chuyên gia, những hệ có khả năng cung cấp "kinh nghiệm điều khiển hệ thống" hay còn gọi là các hệ trợ giúp quyết định. Trí tuệ nhân tạo được xây dựng dựa trên *mạng nơ-ron* nhân tạo. Sự kết hợp giữa logic mờ và mạng nơ-ron trong thiết kế hệ thống điều khiển tự động là một khuynh hướng hoàn toàn mới, phương hướng thiết kế hệ điều khiển thông minh, một hệ thống mà bộ điều khiển có khả năng tư duy như bộ não của con người, tức là nó có khả năng tự học hỏi, tự chỉnh định lại cho phù hợp với sự thay đổi không lường được trước của đối tượng điều khiển. Bài giảng này sẽ trình bày nhanh những nét cơ bản nhất của mạng nơ-ron nhân tạo cũng như các phương pháp thường dùng để huấn luyện cho nó.

1 Mô hình mạng nơ-ron nhân tạo

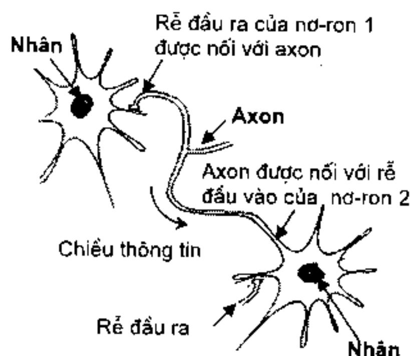
1.1 Cấu trúc một nơ-ron nhân tạo

Mạng nơ-ron là sự tái tạo bằng kỹ thuật những chức năng của hệ thần kinh con người. Trong quá trình tái tạo không phải tất cả các chức năng của bộ não con người có đều được tái tạo, mà chỉ có những chức năng cần thiết. Bên cạnh đó còn có những chức năng mới được tạo ra nhằm giải quyết một bài toán điều khiển đã định hướng trước.

Mạng nơ-ron bao gồm vô số các nơ-ron được liên kết truyền thông với nhau trong mạng. Hình 1 là một phần của mạng nơ-ron bao gồm hai nơ-ron.

Một nơ-ron chứa đựng các thành phần cơ bản:

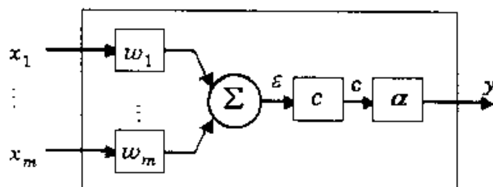
- Thân nơ-ron được giới hạn trong một màng membran và trong cùng là nhân. Từ thân nơ-ron còn có rất nhiều đường rẽ nhánh tạm gọi là rễ.
- "Bus" liên kết nơ-ron này với các nơ-ron khác được gọi là axon, trên axon có các đường rẽ nhánh. Nơ-ron còn có thể liên kết với các nơ-ron khác qua các rễ. Chính vì cách liên kết đa dạng như vậy nên mạng nơ-ron có độ liên kết rất cao.



Hình 1: Một mạng nơ-ron đơn giản gồm hai nơ-ron.

Các rễ của nơ-ron được chia thành hai loại: loại nhận thông tin từ nơ-ron khác qua axon, mà ta sẽ gọi là *rễ đầu vào* và loại đưa thông tin qua axon tới các nơ-ron khác, gọi là *rễ đầu ra*. Một nơ-ron có thể có *nhiều rễ đầu vào, nhưng chỉ có một rễ đầu ra*. Bởi vậy nếu xem nơ-ron như một khâu điều khiển thì nó chính là khâu có nhiều đầu vào, một đầu ra (khâu MISO). Một tính chất rất cơ bản của mạng nơ-ron sinh học là các đáp ứng theo kích thích có khả năng thay đổi theo thời gian. Các đáp ứng có thể tăng lên, giảm đi hoặc hoàn toàn biến mất. Qua các nhánh axon liên kết tế bào nơ-ron này với các nơ-ron khác, sự thay đổi trạng thái của một nơ-ron cũng sẽ kéo theo sự thay đổi trạng thái của những nơ-ron khác và do đó là sự thay đổi của toàn bộ mạng nơ-ron. Việc thay đổi trạng thái của mạng nơ-ron có thể thực hiện qua một quá trình "dạy" hoặc do khả năng "học" tự nhiên.

Sự thay thế những tính chất này bằng một mô hình toán học tương đương được gọi là mạng nơ-ron nhân tạo. Mạng nơ-ron nhân tạo có thể được chế tạo bằng nhiều cách khác nhau vì vậy trong thực tế tồn tại rất nhiều kiểu mạng nơ-ron nhân tạo.



Hình 2: Mô hình một nơ-ron nhân tạo

Hình 2 biểu diễn một nơ-ron nhân tạo gồm m (rễ) đầu vào x_1, … , x_m và một (rễ) đầu ra y. Mô hình này gồm ba thành phần cơ bản:

1) *Khâu cộng*: Các kích thích (đầu vào) của tế bào nơ-ron có thế năng tác động vào màng membran khác nhau được biểu diễn qua các trọng số w_i, $i=1,\ldots,m$ tương ứng với cường độ kích thích của từng đầu vào. Tổng giá trị của các kích thích đầu vào được thực hiện qua một bộ cộng:

$$\varepsilon = \sum_{i=1}^{m} w_i x_i = \underline{w}^T \underline{x} \quad \text{với} \quad \underline{w} = \begin{pmatrix} w_1 \\ \vdots \\ w_m \end{pmatrix}, \ \underline{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix} \qquad (1.1)$$

2) *Khâu tiền đáp ứng*: Đầu ra của khâu cộng được đưa đến khâu tiền đáp ứng c. Một cách đơn giản nhất có thể tạo đáp ứng đầu ra là:

$$c = \varepsilon \qquad (1.2)$$

Nhưng để tăng độ chính xác về đặc tính động học, phù hợp với nguyên lý làm việc của nơ-ron rằng khi có kích thích $\varepsilon(t)$ đầu vào, thế năng $c(t)$ của màng membran tăng dần có quán tính, người ta thay thế (1.2) bằng:

$$T\frac{dc(t)}{dt} + c(t) - c_0 = \varepsilon(t) \qquad (1.3)$$

trong đó c_0 là thế năng của mạng membran ở trạng thái không bị kích thích. Đó là phương trình động học của một khâu quán tính bậc nhất với hằng số thời gian quán tính T. Khâu tạo chức năng đáp ứng kiểu này còn có tên là khâu BSB.

Bên cạnh khâu tạo đáp ứng $c(t)$ kiểu tuyến tính và kiểu BSB còn tồn tại các kiểu khâu theo kiểu gián đoạn. Thuộc nhóm khâu kiểu gián đoạn có khâu tạo đáp ứng theo hàm *Hopfield*:

$$c = \begin{cases} -1 & \text{khi} \quad \varepsilon < 0 \\ +1 & \text{khi} \quad \varepsilon \geq 0 \end{cases} \tag{1.4}$$

3) *Khâu tạo đáp ứng đầu ra*: Khi thế năng $c(t)$ của màng membran vượt quá ngưỡng, nơ-ron sẽ ở trạng thái tích cực:

$$y = \alpha(c) = \begin{cases} 0 & \text{khi} \quad c < c_0 \\ 1 & \text{khi} \quad c \geq c_0 \end{cases} \tag{1.5}$$

Tuy nhiên, sự chuyển đổi trạng thái của nơ-ron từ không tích cực sang tích cực và ngược lại thường là quá trình liên tục. Một trong những khâu α mô tả được quá trình liên tục đó là khâu *Sigma* biểu diễn dưới dạng hàm *Fermi*:

$$y = \alpha(c) = \frac{1}{1 + e^{-c}} \tag{1.6}$$

Ngoài ra người ta còn sử dụng nhiều khâu α khác, chẳng hạn như:

a) Khâu lưỡng tuyến tính $y = \begin{cases} 1 + \mathrm{sgn}(c) & \text{khi} \quad |c| > 1 \\ 1 + c & \text{khi} \quad |c| \leq b \end{cases}$ (1.7)

b) Khâu tuyến tính $y = kc$, với k là hằng số. (1.8)

c) Khâu đồng dạng $y = c$ (1.9)

d) Khâu tạo đáp ứng ngẫu nhiên. Khâu này cho ra giá trị y nhị phân (hoặc bằng 0 hoặc bằng 1) tại đầu ra. Điểm khác biệt so với loại khâu tiền định là hàm mô tả khâu không ở dạng tiền định như (1.5)–(1.9) mà là một hàm thông báo xác suất để có được $y = 1$ tại đầu ra. Một trong đại diện cho mô hình mô tả khâu ngẫu nhiên này là hàm *Boltzman* định nghĩa như sau:

$$p(y=1) = \frac{1}{1 + e^{-\frac{c - c_0}{T}}} \tag{1.10}$$

trong đó c_0 là giá trị ngưỡng và tham số T là một đại lượng vật lý biểu diễn độ nhạy cảm của nơ-ron.

Tất cả các dạng hàm mô tả trên của khâu tạo đáp ứng α đều cho ra đáp ứng y là một tín hiệu nằm trong khoảng từ 0 đến 1. Tuy vậy không bắt buộc ta phải giữ nguyên miền giá trị đó của y. Tùy vào từng bài toán ứng dụng, ta có thể thay đổi chúng sao cho y nhận giá trị chẳng hạn như từ y_{min} đến y_{max} cho trước.

Mỗi một kết nối từ vector tín hiệu vào $\underline{x}$ tới tín hiệu ra y, qua đặc tính của khâu cộng, khâu tiền đáp ứng c và khâu tạo đáp ứng α sẽ cho ra một mô hình nơ-ron. Như vậy tổng cộng sẽ có tất cả là 15 mô hình nơ-ron. Giá trị đầu ra y của nơ-ron là:

$$y = \alpha(c) = \alpha(c(\varepsilon)) = \alpha(c(\underline{w}^T \underline{x})) = \alpha \circ c(\underline{w}^T \underline{x})$$

Tuy nhiên, phổ biến nhất trong số 15 mô hình nơ-ron nêu trên là 6 loại cho trong bảng sau:

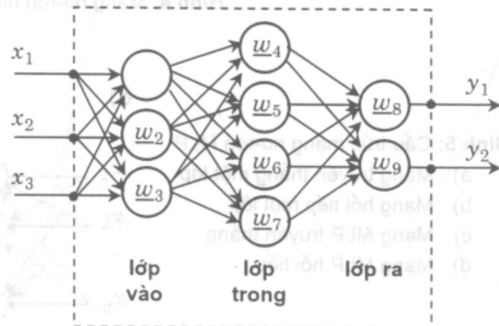
Tên gọi	Khâu c	Khâu α	Tên gọi	Khâu c	Khâu α
McCulloch–Pitts	(1.2)	(1.5)	Adeline	(1.2)	(1.9)
Fermi	(1.2)	(1.6)	Boltzmann	(1.2)	(1.10)
BSB	(1.3)	(1.7)	Hopfield	(1.4)	(1.9)

1.2 Các cấu trúc cơ bản của mạng nơ-ron nhân tạo

Liên kết các đầu vào và ra của nhiều nơ-ron với nhau ta được một *mạng nơ-ron*. Việc ghép nối các nơ-ron trong mạng với nhau có thể theo một nguyên tắc bất kỳ nào đó. Ở một mạng, các nơ-ron được phân biệt với nhau thông qua vị trí của nó trong mạng, cụ thể là

- Nhóm nơ-ron đầu vào (*input layer*) là những nơ-ron nhận thông tin từ môi trường bên ngoài vào trong mạng. Chúng có vị trí ngoài cùng "bên trái" và được nối với các nơ-ron khác trong mạng từ (rễ) đầu ra.
- Nhóm nơ-ron được các nơ-ron khác trong mạng kết nối tới thông qua (rễ) đầu vào được gọi là nơ-ron đầu ra (*output layer*). Những nơ-ron đầu ra có vị trí ngoài cùng "bên phải" và có nhiệm vụ đưa tín hiệu của mạng ra bên ngoài.
- Những nơ-ron còn lại không thuộc hai nhóm trên được gọi là nơ-ron bên trong (*hidden layer*).

Như vậy một mạng nơ-ron cũng có chức năng của một hệ truyền đạt và xử lý tín hiệu từ đầu vào đến đầu ra của mạng. Các nơ-ron trong một mạng thường được chọn cùng một loại, chúng được phân biệt với nhau qua vector hàm trọng số $\underline{w}$.

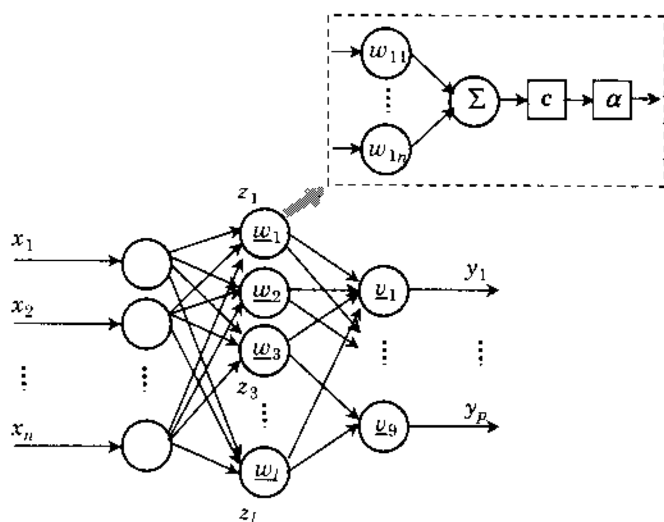


Hình 3: Mạng nơ-ron ba lớp truyền thẳng

Một mạng nơ-ron được chia thành các lớp, mỗi lớp bao gồm nhiều nơ-ron có cùng một chức năng trong mạng. Hình 3 là mô hình của một mạng nơ-ron ba lớp với 9 nơ-ron. Mạng có ba đầu vào x_1, x_2, x_3 và hai đầu ra y_1, y_2. Các tín hiệu đầu vào được đưa đến ba nơ-ron đầu vào, ba nơ-ron này làm thành lớp đầu vào của mạng (*input layer*). Đầu ra của các nơ-ron này được đưa đến đầu vào của bốn nơ-ron tiếp theo, bốn nơ-ron này không trực tiếp tiếp xúc với môi trường xung quanh và làm thành lớp trung gian trong mạng (*hidden layer*). Đầu ra của các nơ-ron này được đưa đến hai nơ-ron đưa tín hiệu ra môi trường bên ngoài, thuộc lớp các nơ-ron đầu ra (*output layer*).

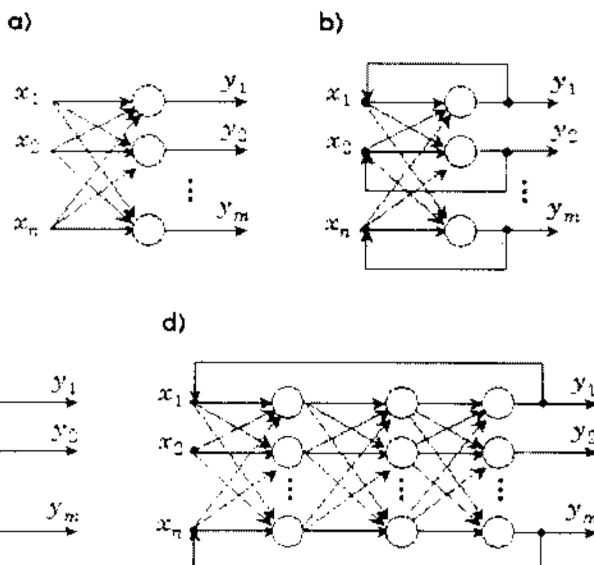
Mạng nơ-ron mà ở đó không tồn tại bất kỳ một mạch hồi tiếp nào kể cả hồi tiếp nội lẫn hồi tiếp từ đầu ra trở về đầu vào, được gọi mạng truyền thẳng (*feedforward network*). Ngược lại, mạng có đường phản hồi từ đầu ra của một nơ-ron tới đầu vào của nơ-ron cùng lớp hoặc thuộc lớp phía trước có tên gọi là mạng hồi tiếp (*feedback network*).

Mạng nơ-ron bao gồm một hay nhiều lớp trong (lớp trung gian) được gọi là mạng **MLP** (*multilayer perceptrons Network*). Còn mạng chỉ có một lớp, vừa là lớp vào vừa là lớp trung gian và cũng là lớp ra thì mạng đó có tên là *mạng một lớp*.



Hình 4: Mạng nơ-ron nhiều lớp (mạng MLP)

Hình 5: Cấu trúc mạng nơ-ron MLP
- a) Mạng truyền thẳng một lớp
- b) Mạng hồi tiếp một lớp
- c) Mạng MLP truyền thẳng
- d) Mạng MLP hồi tiếp



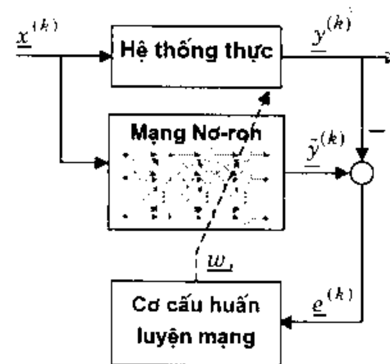
Mạng nơ-ron luôn có cấu trúc *ghép nối hoàn toàn*, tức là bất cứ một nơ-ron nào trong mạng cũng được nối với một hoặc vài nơ-ron khác. Trong trường hợp các nơ-ron trong mạng có khâu tạo chức năng đáp ứng là khâu tuyến tính, tính phi tuyến chỉ nằm ở khâu tạo chức năng ra thì việc mắc nối tiếp các nơ-ron trong mạng không còn ý nghĩa nữa và lúc đó ta hoàn toàn có thể thay thế mạng nơ-ron nhiều lớp thành mạng nơ-ron một lớp.

2 Huấn luyện mạng

2.1 Nguyên tắc huấn luyện mạng

Huấn luyện mạng là công việc xác định các vector trọng số $\underline{w}_i$ có trong từng nơ-ron của mạng sao cho mạng nơ-ron có khả năng tạo ra các đáp ứng đầu ra mong muốn (giống như tín hiệu ra của một hệ thống) khi cùng được kích thích bằng một lượng thông tin đầu vào của hệ thống đó. Như vậy, có thể xem việc huấn luyện mạng là tạo ra cho mạng khả năng của một thiết bị xấp xỉ thông tin.

Để huấn luyện, người ta cho tác động vào mạng hàng loạt kích thích $\underline{x}^{(k)}$, $k=1,2, \ldots$ có khả năng lặp lại trong quá trình mạng làm việc. Những tác động này được gọi là kích thích mẫu. Các giá trị $\underline{\tilde{y}}^{(k)}$ tương ứng tại đầu ra của mạng được so sánh với đáp ứng mẫu $\underline{y}^{(k)}$ cho trước. Các phần tử của những vector trọng số $\underline{w}_i$, $i=1,2, \ldots, n$ có trong tất cả n nơ-ron của mạng được hiệu chỉnh theo từng bước huấn luyện sao cho tổng các sai lệch $E_k = \left|\underline{e}^{(k)}\right|^2 = \left|\underline{\tilde{y}}^{(k)} - \underline{y}^{(k)}\right|^2$ là nhỏ nhất. Như vậy, thực chất việc huấn luyện mạng chính là quá trình giải bài toán tối ưu tham số và để thực hiện được bài toán tối ưu tham số này phải xây dựng được phiếm hàm mục đích mô tả rõ ràng sai lệch. Hình 6 mô tả nguyên lý huấn luyện một mạng nơ-ron.



Hình 6: Nguyên tắc huấn luyện mạng

Để minh họa quy trình huấn luyện mạng nơ-ron, ta quan sát một mạng MLP ba lớp như đã mô tả ở hình 4 gồm các lớp vào có n đầu vào $x_1, x_2, \cdots, x_n$, lớp trong với l nơ-ron và lớp ra với p đầu ra $y_1, y_2, \cdots, y_p$. Các nơ-ron đầu vào chỉ đóng vai trò thuần túy thu thập thông tin tác động vào mạng và không tham gia vào quá trình huấn luyện. Quá trình huấn luyện chỉ thực hiện thông qua các nơ-ron ở lớp trong và ở lớp ra. Ta sẽ ký hiệu phần tử của các vector trọng số $\underline{w}_i$ thuộc nơ-ron thứ i của lớp trong với n phần tử là $w_{i1}, w_{i2}, \cdots, w_{in}$ và vector trọng số $\underline{v}_j$ của nơ-ron ở lớp đầu ra gồm l phần tử là $v_{j1}, v_{j2}, \cdots, v_{jl}$. Đầu ra của các nơ-ron lớp trong có ký hiệu là $z_1, z_2, \cdots, z_l$.

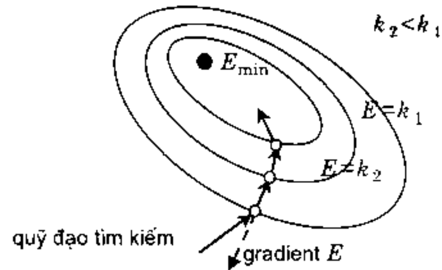
Giả sử, tại thời điểm huấn luyện thứ m ta đã có m cặp vector các giá trị $\{\underline{x}^{(k)}, \underline{y}^{(k)}\}$, $k=1,2,\ \ldots\ m$. Ứng với mỗi vector đầu vào $\underline{x}^{(k)}$ ta có một vector đáp ứng đầu ra $\underline{\tilde{y}}^{(k)}$ của mạng. Sai lệch E_k ứng với mẫu học thứ k được biểu diễn dưới dạng hàm mục đích bằng bình phương sai lệch giữa mẫu $\underline{y}^{(k)}$ và đáp ứng ra thực của mạng $\underline{\tilde{y}}^{(k)}$ có dạng :

$$E_k = \frac{1}{2}\left|\underline{y}^{(k)} - \underline{\tilde{y}}^{(k)}\right|^2 = \frac{1}{2}\sum_{j=1}^{p}\left[y_j^{(k)} - \tilde{y}_j^{(k)}\right]^2$$

Khi đó, tổng tất cả các sai lệch được tính đến thời điểm huấn luyện thứ m sẽ là:

$$E = \sum_{k=1}^{m} E_k = \frac{1}{2}\sum_{k=1}^{m}\sum_{j=1}^{p}\left[y_j^{(k)} - \tilde{y}_j^{(k)}\right]^2$$

Quá trình học là thực hiện nhiệm vụ *tìm giá trị cực tiểu* E_{min} cho hàm mục đích E bằng cách thay đổi giá trị của các phần tử trong vector hàm trọng số $\underline{w}_i$ sao cho sau mỗi lần thay đổi như vậy giá trị của E chẳng hạn đang là k_1 sẽ được giảm đi một ít thành $k_2<k_1$ (hình 7). Thông thường việc giải bài toán tối ưu tĩnh này được thực hiện theo nguyên lý chuyển động đến điểm cực trị.



Hình 7: Minh họa phương pháp tìm kiếm E_{min} theo hướng ngược gradient của E.

Một trong những cách thay đổi vector hàm trọng số nhằm làm giảm giá trị của E là dịch chuyển theo hướng ngược gradient vì vector gradient E là $\left(\frac{\partial E}{\partial \underline{w}}\right)^T$ luôn có hướng chỉ chiều tăng giá trị của E. Nguyên tắc học này có thể diễn đạt dưới dạng xác định lượng hiệu chỉnh như sau:

$$\Delta w_{iq} = -s\frac{\partial E}{\partial w_{iq}} \quad ; \quad i=1,\ 2,\ \ldots\ ,\ l\ ; \quad q=1,\ 2,\ \ldots\ ,\ n$$

cho lớp nơ-ron trong và

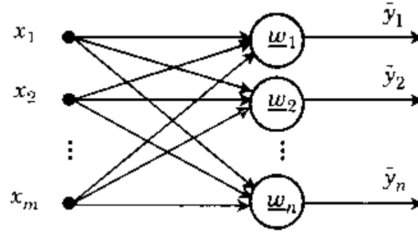
$$\Delta v_{jr} = -s\frac{\partial E}{\partial v_{jr}} \quad ; \quad j=1,\ 2,\ \ldots\ ,\ p\ ; \quad r=1,\ 2,\ \ldots\ ,\ l$$

cho lớp nơ-ron đầu ra, trong đó hằng số s thường có thể được chọn trước trong khoảng từ 0.1 đến 0.9 (có vai trò như khoảng cách bước tìm, càng nhỏ thì nghiệm tìm được càng chính xác, song lại lâu hội tụ – xem thêm tài liệu [6]). Sau đó những giá trị w_{iq}, v_{jr} mới sẽ được xác định từ giá trị w_{iq}, v_{jr} cũ và các lượng hiệu chỉnh Δw_{iq}, Δv_{jr} theo:

$$w_{iq}^{(mới)} = w_{iq}^{(cũ)} + \Delta w_{iq} \quad \text{và} \quad v_{jr}^{(mới)} = v_{jr}^{(cũ)} + \Delta v_{jr}$$

2.2 Huấn luyện mạng truyền thẳng một lớp

Mạng nơ-ron truyền thẳng một lớp là mạng có cấu trúc đơn giản nhất. Tính đơn giản này hoàn toàn không hạn chế miền ứng dụng của nó vì mọi mạng MLP với các nơ-ron thuộc lớp trong (lớp trung gian) và lớp đầu vào có tính tuyến tính đều chuyển được về dạng một lớp. Do có khả năng đại diện tổng quát cho mạng MLP với tính tuyến tính của các nơ-ron lớp trung gian và lớp đầu vào như vậy nên mạng nơ-ron một lớp còn có tên gọi là *mạng tuyến tính*.



Hình 8: Mạng nơ-ron truyền thẳng một lớp với m đầu vào, n đầu ra.

2.2.1 Mạng Adeline

Mạng Adeline (phần tử thích nghi tuyến tính – *Adaptive Linear Element*) là mạng có cấu trúc như ở hình 8 mô tả, với khâu tiền đáp ứng kiểu (1.2) và khâu tạo đáp ứng đầu ra đồng dạng cho bởi (1.9) trong tất cả n nơ-ron, tức là:

$$\tilde{y}_i = \underline{w}_i^T \underline{x} = \sum_{j=1}^{m} w_{ij}x_j \quad \text{trong đó } \underline{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix},\ \tilde{y}_i = \begin{pmatrix} \tilde{y}_1 \\ \vdots \\ \tilde{y}_n \end{pmatrix},\ \underline{w}_i = \begin{pmatrix} w_{i1} \\ \vdots \\ w_{im} \end{pmatrix},\ i=1,2,\ \dots\ ,n$$

Nhiệm vụ huấn luyện mạng là xác định các trọng số w_{ij}, $i=1,2,\ \dots\ ,n$; $j=1,2,\ \dots\ ,m$

để tại bước điều khiển thứ k bất kỳ luôn có đáp ứng đầu ra $\underline{\tilde{y}}^{(k)} = \begin{pmatrix} \tilde{y}_1^{(k)} \\ \vdots \\ \tilde{y}_n^{(k)} \end{pmatrix}$ ứng với tác động

$\underline{x}^{(k)} = \begin{pmatrix} x_1^{(k)} \\ \vdots \\ x_m^{(k)} \end{pmatrix}$ ở đầu vào được "giống như" mẫu $\underline{y}^{(k)} = \begin{pmatrix} y_1^{(k)} \\ \vdots \\ y_n^{(k)} \end{pmatrix}$ mong muốn, tức là để có:

$$E_k = \left|\underline{y}^{(k)} - \underline{\tilde{y}}^{(k)}\right|^2 = \left|\underline{y}^{(k)} - \underline{w}_i^T \underline{x}^{(k)}\right|^2 \to \min$$

$$\Leftrightarrow \quad E_k = \left|\underline{y}^{(k)} - \underbrace{\begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1m} \\ w_{21} & w_{22} & \cdots & w_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nm} \end{pmatrix}}_{W} \underline{x}^{(k)}\right|^2 = \left|\underline{y}^{(k)} - W\underline{x}^{(k)}\right|^2 \to \min \qquad (2.1)$$

Giả sử rằng ta đã có tất cả p mẫu học $\{\underline{x}^{(k)}, \underline{y}^{(k)}\}$, $k=1,\ 2,\ \dots\ ,\ p$. Khi đó điều kiện (2.1) cho tất cả p bước sẽ tương đương với:

$$E = \frac{1}{2}\sum_{k=1}^{p} E_k = \frac{1}{2}\sum_{k=1}^{p}\left[\underline{y}^{(k)} - W\underline{x}^{(k)}\right]^2 \to \min \tag{2.2}$$

và từ đây ta suy ra được với một khoảng cách bước tìm s cố định cho trước:

$$\Delta w_{ij} = -s\frac{\partial E}{\partial w_{ij}} = s\sum_{k=1}^{p}\left[y_i^{(k)} - \underline{w}_i^T\underline{x}^{(k)}\right]x_j^{(k)} \tag{2.3}$$

Nhược điểm của công thức (2.3) để hiệu chỉnh trọng số w_{ij} là phải có ngay một lúc toàn bộ p mẫu học $\{\underline{x}^{(k)}, \underline{y}^{(k)}\}$, $k=1, 2, \ldots, p$. Nhằm khắc phục, Widnow năm 1962 đã đề nghị sử dụng dạng cải biên của nó tại bước học thứ k thành:

$$\Delta w_{ij}^{(k)} = s\left[y_i^{(k)} - \left(\underline{w}_i^T\right)^{(k)}\underline{x}^{(k)}\right]x_j^{(k)} \tag{2.4}$$

trong đó $\left(\underline{w}_i^T\right)^{(k)} = \left(w_{i1}^{(k)}, w_{i2}^{(k)}, \cdots, w_{im}^{(k)}\right)^T$. Với giá trị gia tăng $\Delta w_{ij}^{(k)}$ trên ta sẽ thu được sự hiệu chỉnh trọng số $w_{ij}^{(k+1)}$ tại bước thứ $k+1$ từ các giá trị đã có là $w_{ij}^{(k)}$ ở ngay bước thứ k trước đó một cách đơn giản hơn như sau:

$$w_{ij}^{(k+1)} = w_{ij}^{(k)} + \Delta w_{ij}^{(k)} \tag{2.5}$$

2.2.2 Mạng tuyến tính sử dụng nơ-ron Hopfield

Mạng tuyến tính có cấu trúc như ở hình 8 với khâu tiền đáp ứng Hopfield (1.4) và khâu tạo đáp ứng đầu ra đồng dạng (1.9) trong tất cả n nơ-ron, sẽ có đầu ra:

$$\tilde{y}_i = \text{sgn}(\underline{w}_i^T\underline{x}),\ \text{trong đó } \underline{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_m \end{pmatrix},\ \tilde{y}_i = \begin{pmatrix} \tilde{y}_1 \\ \vdots \\ \tilde{y}_n \end{pmatrix},\ \underline{w}_i = \begin{pmatrix} w_{i1} \\ \vdots \\ w_{im} \end{pmatrix},\ i=1,2,\ldots,n \tag{2.6}$$

Như vậy, giống như ở mạng Adeline, cụ thể là với công thức (2.2), ta có hàm mục tiêu cho quá trình xác định trọng số w_{ij} từ p mẫu học $\{\underline{x}^{(k)}, \underline{y}^{(k)}\}$, $k=1, 2, \ldots, p$ như sau:

$$E = \frac{1}{2}\sum_{k=1}^{p}\left[\underline{y}^{(k)} - \underline{\tilde{y}}^{(k)}\right]^2 = \frac{1}{2}\sum_{k=1}^{p}\left[\underline{y}^{(k)} - \text{sgn}\left(W\underline{x}^{(k)}\right)\right]^2 \to \min \tag{2.7}$$

Do đó nếu gọi s là khoảng cách bước tìm, thì với (2.7) ta có công thức tính $\Delta w_{ij}^{(k)}$ ở bước học thứ k đã được cải biên theo Widnow (2.4) là:

$$\Delta w_{ij}^{(k)} = s\left[y_i^{(k)} - \text{sgn}\left(\left(\underline{w}_i^T\right)^{(k)}\underline{x}^{(k)}\right)\right]x_j^{(k)} = \begin{cases} 2sy_i^{(k)}x_j^{(k)} & \text{nếu } y_i^{(k)} \neq \tilde{y}_i^{(k)} \\ 0 & \text{cho các trường hợp khác} \end{cases} \tag{2.8}$$

vì $y_i^{(k)}$ chỉ có giá trị là -1 hoặc 1, trong đó $\left(\underline{w}_i^T\right)^{(k)} = \left(w_{i1}^{(k)}, w_{i2}^{(k)}, \cdots, w_{im}^{(k)}\right)^T$.

Cuối cùng, với giá trị gia tăng $\Delta w_{ij}^{(k)}$ theo (2.8) ta sẽ thu được sự hiệu chỉnh $w_{ij}^{(k+1)}$ tại bước thứ $k+1$ từ các giá trị đã có là $w_{ij}^{(k)}$ ở ngay bước thứ k trước đó như sau:

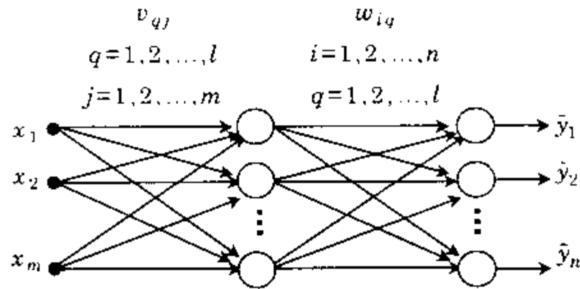
$$w_{ij}^{(k+1)} = w_{ij}^{(k)} + \Delta w_{ij}^{(k)} \tag{2.9}$$

2.3 Huấn luyện mạng MLP truyền thẳng

2.3.1 Nguyên tắc huấn luyện chung

Khi huấn luyện mạng một lớp với mẫu học $\{\underline{x}^{(k)}, \underline{y}^{(k)}\}$, $k=1, 2, \ldots$ cho trước mà thuật toán xác định vector trọng số không hội tụ, người ta phải nghĩ tới mạng nhiều lớp (MLP) và đặc biệt mạng MLP này phải có quan hệ phi tuyến không chỉ nằm riêng ở lớp nơ-ron đầu ra.

Mạng MLP có cấu trúc đơn giản nhất là mạng truyền thẳng. Hình 9 mô tả một mạng nơ-ron MLP hai lớp truyền thẳng với m đầu vào x_j, $j=1,2,\ldots,m$ và n đầu ra $\tilde{y}_i$, $i=1,2,\ldots,n$. Lớp nơ-ron thứ nhất có l nơ-ron với l đầu ra $z_1, z_2, \ldots, z_l$, tức là có:



Hình 9: Mạng nơ-ron hai lớp truyền thẳng

$$c_q = \sum_{j=1}^{m} v_{qj} x_j \qquad \Rightarrow \qquad z_q = \alpha(c_q) = \alpha\left(\sum_{j=1}^{m} v_{qj} x_j\right)$$

Lớp nơ-ron thứ hai có n nơ-ron với n đầu ra $\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_n$:

$$c_i = \sum_{q=1}^{l} w_{iq} z_q \qquad \Rightarrow \qquad \tilde{y}_i = \alpha(c_i) = \alpha\left(\sum_{q=1}^{l} w_{iq} z_q\right)$$

Suy ra

$$\tilde{y}_i = \alpha\left(\sum_{q=1}^{l} w_{iq} \alpha\left(\sum_{j=1}^{m} v_{qj} x_j\right)\right) \tag{2.10}$$

Nhiệm vụ của việc dạy học cho mạng trên là xác định các trọng số w_{iq}, v_{qj} từ mẫu học $\{\underline{x}^{(k)}, \underline{y}^{(k)}\}$, $k=1, 2, \ldots$ sao cho:

$$E = \frac{1}{2}\sum_{(k)}\sum_{i=1}^{n}\left[y_i^{(k)} - \tilde{y}_i^{(k)}\right]^2 \rightarrow \min \tag{2.11}$$

Nguyên tắc chung để xác định w_{iq}, v_{qj} theo hướng ngược gradient của E đã được trình bày ở hình 7 ngay đầu mục 2 này. Gọi $w_{iq}^{(cũ)}$, $v_{qj}^{(cũ)}$ là giá trị các trọng số đã có. Những giá trị này sẽ được hiệu chỉnh bằng một đại lượng gia tăng Δw_{iq}, Δv_{qj} để có:

$$w_{iq}^{(mới)} = w_{iq}^{(cũ)} + \Delta w_{iq} \qquad \text{và} \qquad v_{qj}^{(mới)} = v_{qj}^{(cũ)} + \Delta v_{qj} \tag{2.12}$$

trong đó Δw_{iq}, Δv_{qj} được tính theo hướng ngược gradient:

$$\Delta w_{iq} = -s\frac{\partial E}{\partial w_{iq}} = -s\frac{\partial E}{\partial \tilde{y}_i}\frac{\partial \tilde{y}_i}{\partial c_i}\frac{\partial c_i}{\partial w_{iq}} = s\sum_{(k)}\left[y_i^{(k)} - \tilde{y}_i^{(k)}\right]\left.\frac{d\alpha}{dc_i}\right|_{c_i^{(k)}} z_q \tag{2.13}$$

$$\Delta v_{qj} = -s\frac{\partial E}{\partial v_{qj}} = -s\frac{\partial E}{\partial c_q}\frac{\partial c_q}{\partial v_{qj}} = s\sum_{(k)}\sum_{i=1}^{n}\left[\left(y_i^{(k)} - \tilde{y}_i^{(k)}\right)\frac{d\alpha}{dc_i}w_{iq}^{(cũ)}\right]\left.\frac{d\alpha}{dc_q}\right|_{c_q^{(k)}} x_j \tag{2.14}$$

Tuy nhiên phương pháp trên lại đòi hỏi phải có ngay một lúc toàn bộ mẫu học mà điều này là không thể. Một phương pháp khác được ứng dụng nhiều trong thực tế để tìm kiếm Δw_{iq}, Δv_{qj} là thuật toán lan truyền ngược (*back propagation*) sẽ được trình bày sau đây.

2.3.2 Thuật toán lan truyền ngược

Lan truyền ngược là thuật toán được dùng phổ thông nhất trong việc dạy mạng nhiều lớp, kể cả mạng với nơ-ron có tính động học BSB. Từ một mẫu học cụ thể $\underline{x}^{(k)}$, $\underline{y}^{(k)}$ và các trọng số đã có của mạng, chẳng hạn như $\underline{w}^{(k)}$, $\underline{v}^{(k)}$ ở mạng hai lớp, người ta xác định đầu ra thực $\underline{\tilde{y}}^{(k)}$. Sau đó trên cơ sở so sánh với mẫu học $\underline{y}^{(k)}$, các trọng số của lớp nơ-ron đầu ra, ví dụ $\underline{w}^{(k)}$, được hiệu chỉnh thành $\underline{w}^{(k+1)}$. Tiếp tục, từ trọng số mới $\underline{w}^{(k+1)}$ người ta lại hiệu chỉnh trọng số của các nơ-ron thuộc lớp phía trước, ví dụ như $\underline{v}^{(k)}$ thành $\underline{v}^{(k+1)}$. Cứ như vậy cho đến trọng số của lớp nơ-ron đầu vào.

2.3.2.1 Minh họa thuật toán với mạng 2 lớp

Để phần giải thích chi tiết thuật toán lan truyền ngược được đơn giản, sau đây ta sẽ lấy mạng hai lớp ở hình 9 làm ví dụ. Với sai lệch cho riêng mẫu học thứ k là $\underline{y}^{(k)} - \underline{\tilde{y}}^{(k)}$, giá trị gia tăng $\Delta w_{iq}^{(k)}$ được xác định theo công thức cải biên của Widnow từ (2.13) như sau:

$$\Delta w_{iq}^{(k)} = \underbrace{s\left[y_i^{(k)} - \tilde{y}_i^{(k)}\right]\left.\frac{d\alpha}{dc_i}\right|_{c_i^{(k)}}}_{\delta_{oi}} z_q = s\,\delta_{oi} z_q \tag{2.15}$$

trong đó hằng số δ_{oi} có tên gọi tín hiệu sai lệch của nơ-ron đầu ra thứ i. Rõ ràng $\Delta w_{iq}^{(k)}$ phụ thuộc vào z_q. Để tính z_q ta sử dụng các trọng số cũ hiện có của mạng là $\underline{v}^{(k)}$ như sau:

$$z_q = \alpha(c_q^{(k)}) = \alpha\left(\sum_{j=1}^{m} v_{qj}^{(k)} x_j^{(k)}\right) \quad (2.16)$$

Cùng với $\Delta w_{iq}^{(k)}$, trọng số cũ $w_{iq}^{(k)}$ được hiệu chỉnh thành:

$$w_{iq}^{(k+1)} = w_{iq}^{(k)} + \Delta w_{iq}^{(k)} \quad (2.17)$$

Sau khi đã có $w_{iq}^{(k+1)}$, ta xác định giá trị gia tăng $\Delta v_{qj}^{(k)}$ cho trọng số cũ $v_{qj}^{(k)}$ của nơ-ron thuộc lớp đầu vào trước đó nhờ công thức (2.14) nhưng đã được cải biên theo tư tưởng Widnow thành:

$$\Delta v_{qj}^{(k)} = s\underbrace{\sum_{i=1}^{n}\left[\left(y_i^{(k)} - \tilde{y}_i^{(k)}\right)\frac{d\alpha}{dc_i} w_{iq}^{(k+1)}\right]\frac{d\alpha}{dc_q}}_{\delta_{hq}} x_j = s\,\delta_{hq} x_j \quad (2.18)$$

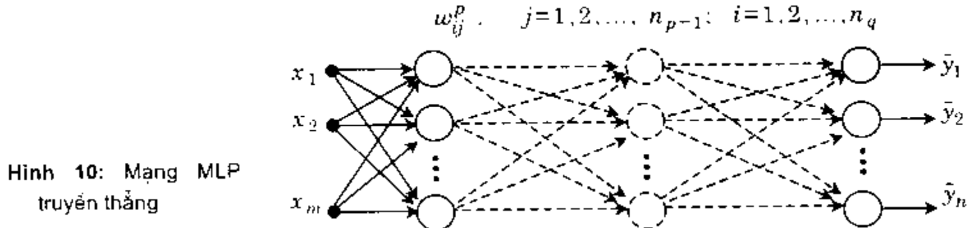
trong đó

$$\delta_{hq} = \sum_{i=1}^{n}\left[\left(y_i^{(k)} - \tilde{y}_i^{(k)}\right)\frac{d\alpha}{dc_i} w_{iq}^{(k+1)}\right]\left.\frac{d\alpha}{dc_q}\right|_{c_q^{(k)}} = \left.\frac{d\alpha}{dc_q}\right|_{c_q^{(k)}} \sum_{i=1}^{n} \delta_{oi} w_{iq}^{(k+1)} \quad (2.19)$$

Từ $\Delta v_{qj}^{(k)}$ ta lại được

$$v_{qj}^{(k+1)} = v_{qj}^{(k)} + \Delta v_{qj}^{(k)} \quad (2.20)$$

2.3.2.2 Dạng thuật toán cho mạng nhiều lớp



Hình 10: Mạng MLP truyền thẳng

Bây giờ ta xét tổng quát mạng MLP truyền thẳng với m đầu vào x_j, $j = 1,2,\ldots,m$ và n đầu ra $\tilde{y}_i$, $i = 1,2,\ldots,n$ gồm Q lớp với mỗi lớp có n_p nơ-ron, $p = 1,2,\ldots,Q$ như hình 10 mô tả. Gọi c_i^p, z_i^p, $p = 1,2,\ldots,Q$ là đối số hàm $\alpha(\cdot)$ cũng như đầu ra của nơ-ron thứ i thuộc lớp thứ p trong mạng. Ký hiệu w_{ij}^p, $j = 1,2,\ldots,n_{p-1}$; $i = 1,2,\ldots,n_p$ là trọng số chuyển z_j^{p-1} thành z_i^p. Như vậy thì:

$$z_i^p = \alpha(\,c_i^p\,) = \alpha\underbrace{\left(\sum_{j=1}^{m_p} w_{ij}^p z_j^{p-1}\right)}_{c_i^p} \tag{2.21}$$

$$z_j^0 = x_j\ ,\quad n_0 = m\ ,\quad z_i^Q = \hat{y}_i \qquad \text{và} \qquad n_Q = n \tag{2.22}$$

Giả sử mạng đang có các trọng số $w_{ij}^{p,(k)}$ và một mẫu học $\underline{x}^{(k)}$, $\underline{y}^{(k)}$. Thuật toán lan truyền ngược xác định trọng số mới $w_{ij}^{p,(k+1)}$ cho mạng MLP như trên sẽ gồm:

- Đặt $z_j^0 = x_j$, $n_0 = m$ và $n_Q = n$
- Tính $c_i^{p,(k)}$ và z_i^p lần lượt cho $p = 1, 2, \ldots, Q$ bằng cách ứng với mỗi giá trị p ta lại thực hiện lần lượt cho $i = 1, 2, \ldots, n_p$ theo:

  $$c_i^{p,(k)} = \sum_{j=1}^{n_{p-1}} w_{ij}^{p,(k)} z_j^{p-1} \qquad \text{và} \qquad z_i^p = \alpha(\,c_i^{p,(k)}\,)$$

- Tính $\delta_{Qi} = \left[\,y_i^{(k)} - z_i^Q\,\right]\left.\dfrac{d\alpha}{dc_i^Q}\right|_{c_i^{Q,(k)}}$ với $i = 1,\ 2,\ \ldots,\ n$
- Thực hiện ngược hướng lần lượt theo $p = Q, Q-1, \ldots, 1$ các bước:

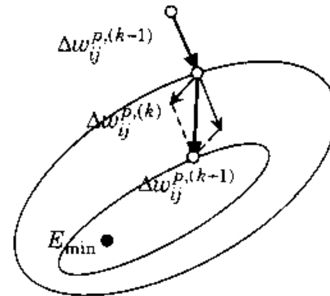
  a) $w_{ij}^{p,(k+1)} = w_{ij}^{p,(k)} + s\,\delta_{pi}\,z_j^{p-1}$ với $j = 1, 2, \ldots, n_{p-1}$ và $i = 1, 2, \ldots, n_p$

  b) Nếu $p > 1$ thì tính:

  $$\delta_{p-1,i} = \left.\frac{d\alpha}{dc_i^{p-1}}\right|_{c_i^{p-1,(k)}} \sum_{j=1}^{n_p} \delta_{pj} w_{ij}^{p,(k+1)} \qquad \text{với} \qquad i = 1, 2, \ldots, n_{p-1}$$

2.3.2.3 Hệ số chỉnh hướng học (momentum)

Để tiếp cận đơn giản tới ý tưởng đưa thêm hệ số chỉnh hướng học vào thuật toán ta lại quay về mạng hai lớp đã mô tả ở hình 9, cụ thể là hướng học tổng quát (2.13) ngược gradient và hướng học đã được cải biên (2.15).



Hình 11: Nguyên lý chỉnh hướng học nhờ hệ số momentum.

Qua so sánh hai hướng học đó ta thấy hướng học (2.15) được sử dụng trong thuật toán có một sai lệch đáng kể so với hướng ngược gradient (2.13) do đã chỉ giữ lại số hạng cuối cùng và bỏ qua tất cả các số hạng còn lại. Bởi vậy, tổng quát

cho mạng nhiều lớp truyền thẳng thì để có được một hướng học mới gần sát (2.13) hơn nữa người ta đã sử dụng $\Delta w_{ij}^{p,(k+1)}$ cho nơ-ron thứ i thuộc lớp thứ p theo công thức:

$$\Delta w_{ij}^{p,(k+1)} = \mu\,\Delta w_{ij}^{p,(k-1)} + \Delta w_{ij}^{p,(k)} = \mu\,\Delta w_{ij}^{p,(k-1)} + s\,\delta_{pi}\,z_j^{p-1}\,,$$

trong đó $\mu \in [0,1]$ được gọi là hệ số chỉnh hướng học (hình 11).

Tài liệu tham khảo

[1] **Chen, C.H. (Ed.)**: Fuzzy Logic and Neural Network Handbook,McGraw-Hill, NewYork. 1996.

[2] **Dubois, D. and Prade, H.**: Fuzzy Sets and Systems, Academic Press, N.Y. . 1980.

[3] **Kahler, J.**: Fuzzy control für Ingenieure. Vieweg Verlag Wiesbaden ,1995.

[4] **Kosko, B.**: Neural Networks and Fuzzy Systems.Prentice-Hall, NJ,1992.

[5] **Minh, P.X. và Phước, N.D.**: Lý thuyết điều khiển mờ, (in lần 4). NXB Khoa học và Kỹ thuật, 2004, Hà nội.

[6] **Phước, N.D.**: Lý thuyết điều khiển nâng cao. NXB Khoa học và Kỹ thuật, 2005, Hà nội.

[7] **Wang, L.X.**: A Course in Fuzzy System and Control. Prentice-Hall International 1997

7 LÝ THUYẾT KHẢ NĂNG VÀ MỘT SỐ VẤN ĐỀ MỞ

Đỗ Văn Thành
Văn phòng Chính phủ

Trình bầy tóm tắt một số nội dung

- Một số khái niệm ban đầu.
- Ngôn ngữ PL1, một số nội dung cơ bản nhất trong PL1.
- Ngôn ngữ PL2, một số vấn đề mở.

1 Một số khái niệm ban đầu

Trong các lý thuyết lập luận xấp xỉ được nghiên cứu và ứng dụng nhiều trong tin học để biểu diễn các thông tin không chắc chắn, được biết không chính xác và không đầy đủ những năm qua như: modal logic, lý thuyết niềm tin và chứng cớ, lý thuyết tập mờ, lý thuyết xác suất, lô gic thời gian, lập luận tương tự, ... thì hiện nay lý thuyết khả năng được nhiều nhà Tin học coi là một trong những lý thuyết có nhiều triển vọng nhất [1] được lựa chọn để xây dựng các hệ chuyên gia, hệ trợ giúp quyết định trên những thông tin được biết không chắc chắn, không đầy đủ.

Lý thuyết khả năng nguyên thuỷ được bắt nguồn từ lý thuyết mờ [7-8], và được D.Dubois, H.Prade phát triển mạnh [2-5]. Nó mang nặng dấu ấn của lý thuyết tập mờ [7]. Nhưng những phát triển về sau này của nó cho thấy lý thuyết này là một sự lai tạo giữa lý thuyết tập mờ và lý thuyết xác suất, nó vừa có mô hình toán học đẹp đẽ của lý thuyết xác suất, vừa có sự mềm dẻo của lý thuyết tập mờ. Lý thuyết khả năng đặc biệt thích hợp để xử lý những thông tin chỉ được biết một phần, không được biết đầy đủ, hoặc mâu thuẫn từng phần. Lý thuyết khả năng có 2 loại: định lượng và định tính. Lý thuyết khả năng định lượng xuất phát từ lý thuyết tập mờ, với ngữ nghĩa dựa trên các phân bố khả năng định lượng [2, 17] (định giá là số thực trong $[0,1]$) trong khi lý thuyết khả năng định tính xuất phát từ Modal logic có ngữ nghĩa dựa trên các quan hệ thự tự yếu (được coi là phân bố khả năng định tính)[1, 4]. Giữa hai loại lý thuyết khả năng này có nhiều mối quan hệ mật thiết với nhau.. Trong trình bầy này chỉ xin phác hoạ một số nét cơ bản của Lô gic khả năng định lượng được phát triển bởi D.Dubois và H.Prade xét trên các ngôn ngữ cổ điển (ngôn ngữ mệnh đề hoặc ngôn ngữ tân từ cấp 1) [3, 5]. Ngữ nghĩa của lô gic này là khá tương tự với ngữ nghĩa lô gic xác suất của Nilsson [6].

Giả sử **L** là ngôn ngữ cổ điển (mệnh đề hoặc tân từ cấp 1). **Ω** là tập các thể hiện đối với các câu trong ngôn ngữ **L**. Một độ đo khả năng định lượng là một ánh xạ Π từ **L** vào [0,1] sao cho

1) $\Pi(\Theta) = 0$; Θ là phần tử (câu) luôn không thể xẩy ra.
2) $\Pi(I) = 1$; I là phần tử luôn xẩy ra.
3) $\Pi(a \vee b) = \max(\Pi(a), \Pi(b))$ với mọi $a, b \in$ **L**.

Một phân bố khả năng định lượng là một ánh xạ $\pi : \Omega \to [0,1]$. π được gọi là phân bố khả năng được chuẩn hoá nếu tồn tại ít nhất một phần tử x^* trong Ω sao cho $\pi(x^*) = 1$.

Khi biết độ đo khả năng thì hoàn toàn xác định được một phân bố khả năng tương ứng. là phân bố chuẩn. và ngược lại. Mối liên hệ giữa phân bố khả năng và độ đo khả năng được thể hiện qua công thức.

$\Pi(S) = \text{suf}\{\pi(w^*)/w \models S\}, b \in \Omega$.

Độ đo cần thiết N là ánh xạ từ **L** vào $[0,1]$ được xác định như sau:

$N(S) = 1 - \Pi(\neg S)$.

Ví dụ: Giả sử ta có các mệnh đề sau:

$S1$: Nếu Lea là sinh viên thì khá chắc chắn rằng cô ta còn trẻ.

$S2$: Nếu Lea còn trẻ thì tương đối chắc chắn rằng cô ta chưa có chồng;

$S3$: Nếu Lea vừa là sinh viên vừa là người mẹ thì gần như chắc chắn rằng cô ta đã có chồng;

$S4$: Chắc chắn rằng Lea là sinh viên;

$S5$: Gần như chắc chắn rằng Lea là người mẹ.

Đặt:

a =" Lea là sinh viên ";
b =" Lea còn trẻ";
c = " Lea chưa có chồng";
d =" Lea là người mẹ".

Lúc đó ta có thể viết lại các mệnh đề trên ở dạng:

$S1: \neg a \vee b$; $S2: \neg b \vee c$; $S3: \neg a \vee \neg d \vee \neg c$; $S4: a$; $S5: d$;

Khi đó tập các thể hiện của ngôn ngữ sinh bởi các câu S_i, $i = 1, \ldots, 5$ sẽ gồm:

$\omega_1 = (a, b, c, d)$; $\omega_2 = (a, b, c, \neg d)$; $\omega_3 = (a, b, \neg c, d)$;

$\omega_4 = (a, b, \neg c, \neg d)$; $\omega_5 = (a, \neg b, c, d)$; $\omega_6 = (a, \neg b, c, \neg d)$;

$\omega_7=(a,\neg b,\neg c,d)$; $\omega_8=(a,\neg b,\neg c,\neg d)$; $\omega_9=(\neg a,b,c,d)$;

$\omega_{10}=(\neg a,b,c,\neg d)$; $\omega_{11}=(\neg a,b,\neg c,d)$; $\omega_{12}=(\neg a,b,\neg c,\neg d)$;

$\omega_{13}=(\neg a,\neg b,c,d)$; $\omega_{14}=(\neg a,\neg b,c,\neg d)$; $\omega_{15}=(\neg a,\neg b,\neg c,d)$;

$\omega_{16}=(\neg a,\neg b,\neg c,\neg d)$;

Nếu tập thể hiện là hữu hạn thì tập này được gọi là tập các thế giới có thể.

Từ định nghĩa trên ta có một số nhận xét sau:

1) Từ $\Pi(S)$ không thể suy ra được $\Pi(\neg S)$, và cũng đúng như vậy đối với độ đo cần thiết N.
2) $\Pi(S) = \inf\{1-\pi(w)/w|=S\}$, với , $w\in\Omega$..
3) $\Pi(a\wedge b)=\min(\Pi(a),\Pi(b))$
4) $\max(\Pi(S),\Pi(\neg S))=1$; $\min(N(S),N(\neg S))=0$.
5) $\Pi(S)\geq N(S)$.
6) $N(S)>0$ thì $\Pi(S)=1$; $\Pi(S)<1$ thì $N(S)=0$.
7) Với câu S bất kỳ, chỉ có 2 tình huống đáng quan tâm có thể xảy ra là $N(S)\geq\beta$ hoặc $\Pi(S)\leq\alpha$.
8) $\Pi(S)+\Pi(\neg S)\geq 1$; $N(S)+N(\neg S)\leq 1$.
9) Độ đo cần thiết được định nghĩa thông qua độ đo khả năng là tương đương với nó cần thoả mãn 3 tiên đề sau:
   a) $N(\Theta)=0$;
   b) $N(I)=1$;
   c) $N(a\wedge b)=\min(N(a),N(b))$ với mọi $a,b\in\mathbf{L}$.

2 Ngôn ngữ PL1

Ký hiệu

$$\mathcal{L}=\{S_i,i=1,2,\ \ldots\ ,n\}$$

là tập các mệnh đề. Khi đó

$$\mathbf{F}=\{(S_i,a_i)\,|\,N(S_i)\geq\alpha_i,\ \alpha_i\in[0,1]\ ,\ i=1,2,\ \ldots\ ,n\}$$

ở đây N là một độ đo cần thiết được gọi là cơ sở tri thức cần thiết giá trị khoảng.

Đối với mỗi cơ sở tri thức cần thiết **F**, nói chung có thể xác định được rất nhiều phân bố khả năng để thoả cơ sở tri thức đã cho. D.Dubois và H.Prade đề xuất công thức xây dựng phân bố khả năng đặc biệt thoả cơ sở tri thức này. Ký hiệu phân bố đó là $\pi_{\mathbf{F}}$ và nó được xác định như sau:

$$\pi_F(\omega) = \min\begin{cases} 1-\alpha_i & \text{nếu} \quad \omega \models \neg S_i \\ 1 & \text{nếu} \quad \omega \models S_1 \wedge S_2 \wedge \cdots \wedge S_n \end{cases}$$

với mọi $\omega \in \Omega$, ([7,8], D.Dubois, H.Prade, 1987, 1994) khi đó độ đo cần thiết N_F xác định từ phân bố này sẽ thoả: $N_F(S_i) \geq \alpha_i$ với mọi $i = 1, 2, \ldots, n$.

2.1 Vấn đề suy diễn trong lý thuyết khả năng PL1

Giả sử **F** là cơ sở tri thức khả năng, $\mathbf{F} = \{(S_i, \alpha_i) \mid \alpha_i \in [0,1], i = 1, 2, \ldots, n\}$, S_i là một công thức lô gic đóng trong ngôn ngữ cổ điển. Phân bố π được gọi là mô hình của (S, α) (ký hiệu $\pi \models (S, \alpha)$) nếu và chỉ nếu $N_\pi(S) \geq \alpha$. Phân bố π được gọi là mô hình của tập các công thức trong **F** ($\pi \models \mathbf{F}$) nếu và chỉ nếu $\pi \models (S_i, \alpha_i)$. (S, α) được gọi là hệ quả lô gic của **F** (ký hiệu $\mathbf{F} \models (S, \alpha)$) nếu với mọi phân bố $\pi / (\pi \models \mathbf{F})$ thì $\pi \models (S, \alpha)$.

Vấn đề suy diễn trong trong ngôn ngữ PL1 được hiểu như sau: Cho cơ sở tri thức khả năng giá trị cần thiết **F**, với S là một câu bất kỳ, ta cần tìm biên dưới tốt nhất của mức độ cần thiết

$$\text{Val}(S, \mathbf{F}) = \sup\{\alpha \mid \mathbf{F} \models (S, \alpha)\}.$$

Cho cơ sở tri thức khả năng **F**, phân bố π_F là phân bố đặc biệt ở chỗ:

Định lý 1: với bất kỳ phân bố khả năng π thoả cơ sở tri thức **F** đã cho thì $\pi \leq \pi_F$, nghĩa là $\pi(\omega) \leq \pi_F(\omega)$ với mọi $\omega \in \Omega$.

Phân bố π_F thực chất được xây dựng theo của nguyên lý đặc tả cực tiểu, mà bản chất của nguyên lý này được bắt nguồn từ ý tưởng của nguyên lý Entropy tối đại. Trong một nghiên cứu của mình [7], chúng tôi đã chỉ ra rằng trong một số điều kiện nào đó thì nguyên lý đặc tả cực tiểu và nguyên lý Entropy tối đại thực sự có quan hệ với nhau, một cách cụ thể là trong những điều kiện như vậy việc áp dụng nguyên lý đặc tả cực tiểu trên các phân bố khả năng là hoàn toàn tương tự như việc áp dụng nguyên lý Entropy tối đại trên các phân bố xác suất được xác định thông qua những phép biến đổi thoả nguyên lý phi mâu thuẫn theo nghĩa của D.Dubois và H.Prade để chuyển việc biểu diễn thông tin từ mô hình của lô gic khả năng sang mô hình của lô gic xác suất và ngược lại.

Lúc đó ta cũng nói π_F là phân bố khả năng đặc trưng của cơ sở tri thức **F**. D.Dubois và H.Prade đã chỉ ra rằng π_F là phân bố được chuẩn hoá khi và chỉ khi $\{S_i, i = 1, 2, \ldots, n\}$ là phi mâu thuẫn trong ngôn ngữ cổ điển.

Lý thuyết khả năng PL1 được phát triển dựa trên phân bố khả năng đặc biệt này.

Định lý 2:

a) $\text{Val}(S, \mathbf{F}) = N_{\mathbf{F}}(S)$.

b) $\mathbf{F} \models (S, \alpha)$ khi và chỉ khi $\pi \models (S, \alpha)$.

2.2 Mâu thuẫn từng phần và suy diễn trong các CSTT mẫu thuẫn từng phần

Định nghĩa: CONS(**F**) = $\sup\pi(\omega)$ được gọi là mức độ phi mâu thuẫn của cơ sở tri thức **F**.

INCONS(**F**) = 1– CONS(**F**) được gọi là mức độ mâu thuẫn của **F**.

$\pi \models \mathbf{F}$, $\omega \in \Omega$.

Định lý 3: INCONS(**F**) = $N_{\mathbf{F}}(\Theta) = \sup\{\alpha \mid \mathbf{F} \models (\Theta, \alpha)\}$.

Giả sử $\mathbf{F}^{\wedge} \subset \mathbf{F}$. sao cho INCONS($\mathbf{F}^{\wedge}$)= INCONS(**F**) và với mọi $\theta = (\varphi, \alpha) \in \mathbf{F}^{\wedge}$. INCONS($\mathbf{F}^{\wedge} - \{\theta\}$) = 0 thì $\mathbf{F}^{\wedge}$ được tập con mâu thuẫn cực tiểu của **F**.

Định lý 4: Nếu INCONS(**F**) = $\alpha > 0$. khi đó tồn tại ít nhất một công thức $(\varphi, \alpha) \in \mathbf{F}^{\wedge}$ và mọi $(\psi, \beta) \in \mathbf{F}^{\wedge}$ thì $\beta \geq \alpha$.

Định lý 5 (về suy diễn): $\mathbf{F} \cup \{(\varphi, 1)\} \models (\psi, \beta)$ nếu và chỉ nếu $\mathbf{F} \models (\varphi \rightarrow \psi, \beta)$.

Định lý 6 (về loại trừ): $\mathbf{F} \models (\varphi, \beta)$ nếu và chỉ nếu $\mathbf{F} \cup \{(\neg\varphi, 1)\} \models (\Theta, \beta)$ và điều này tương đương với: $\text{Val}(\varphi, \mathbf{F}) = \text{INCONS}(\mathbf{F} \cup \{(\neg\varphi, 1)\})$.

Nhận xét: Vấn đề suy diễn được đưa về kiểm tra xem một công thức khả năng có phải là hệ quả lô gic của cơ sở tri thức khả năng hay không?, việc kiểm tra này lại được thực hiện bằng việc tính toán mức độ mâu thuẫn của một cơ sở tri thức khả năng được xác định.

Phân loại tập các công thức **F** như sau:

- **F** hoàn toàn phi mâu thuẫn nếu INCONS(**F**) =0.
- **F** hoàn toàn mâu thuẫn nếu INCONS(**F**) =1.
- **F** mâu thuẫn từng phần nếu 0< INCONS(**F**) <1.

a) CONS(**F**) = $\sup\pi_{\mathbf{F}}(\omega)$ và nếu tồn tại ω^* sao cho $\pi_{\mathbf{F}}(\omega^*) = 1$ thì ω^* là thế giới tốt nhất của **F**.

b) Các trạng thái suy diễn không đơn điệu.

Nhận xét: $\mathbf{F} \models (\varphi, \alpha)$ thì $\mathbf{F} \cup \mathbf{F}' \models (\varphi, \alpha)$, ở đây **F**, **F**' là tập các công thức khả năng bất kỳ.

Ta nói $\mathbf{F} \models (\varphi, \alpha)$ là không tầm thường nếu $\alpha >$ INCONS(**F**). Ký hiệu toán tử suy diễn không tầm thường là |~~, thì **F** |~~ (φ, α) khi và chỉ khi $\mathbf{F} \models (\varphi, \alpha)$ và $\alpha >$ INCONS(**F**).

Định lý 7: Suy diễn $|\sim$ là suy diễn không đơn điệu.

Tức là: có $\mathbf{F} |\sim (\varphi, \alpha)$ thì chưa chắc đã có $\mathbf{F} \cup \mathbf{F}' |\sim (\varphi, \alpha)$.

2.3 Hệ thống hình thức và suy diễn tự động

Mục đích là xây dựng hệ thống các tiên đề và các luật suy diễn sao cho công thức khả năng (φ, α) là hệ quả lô gic của cơ sở tri thức **F** nếu và chỉ nếu nó suy diễn từ **F** trong hệ thống hình thức này.

Hệ tiên đề:

1) $(\varphi \to (\psi \to \varphi), 1)$.
2) $((\varphi \to (\psi \to \xi)) \to ((\varphi \to \psi) \to (\varphi \to \xi)), 1)$.
3) $((\neg \varphi \to \neg \psi) \to ((\neg \varphi \to \psi) \to \varphi), 1)$.
4) $(\forall x (\varphi \to \psi)) \to ((\varphi \to (\forall x \psi), 1)$ nếu x không xuất hiện trong φ và không là biên của ψ.
5) $(\forall x \varphi) \to \varphi_{x/t}, 1)$ nếu x là tự do đối với t trong φ.
   a) GMP: $(\varphi, \alpha), (\varphi \to \psi, \beta) \mid- (\psi, \min(\alpha, \beta))$.
   b) S: $(\varphi, \alpha) \mid- (\varphi, \beta)$ nếu $\alpha \geq \beta$.
   c) G: $(\varphi, \alpha) \mid- (\forall x \varphi, \alpha)$ nếu $\alpha \geq \beta$ nếu x không là biên của φ.

Định lý 8 (Tính hợp lý và tính đầy đủ): $\mathbf{F} \models (S, \alpha)$ nếu và chỉ nếu $\mathbf{F} \vdash (S, \alpha)$.

Như vậy lô gic khả năng giá trị cần thiết đã được tiên đề hoá.

Một mệnh đề khả năng có dạng là một công thức khả năng (c, α), ở đây c là một mệnh đề trong ngôn ngữ cổ điển. Một dạng mệnh đề khả năng là một tuyển hữu hạn mệnh đề khả năng.

$\mathbf{F} = \{(S_i, a_i) /, \ a_i \in [0, 1], \ i = 1, \ \ldots \ , n\}$, mỗi $S_i = \wedge c_{ij}$, c_{ij} là tuyển hữu hạn của các mệnh đề cơ sở trong ngôn ngữ cổ điển.

Ký hiệu $\mathbf{C} = \{(c_{ij}, \ a_i) /, \ a_i \in [0, 1], \ i = 1, \ \ldots \ , n\}$. **C** là tập các mệnh đề giá trị cần thiết của **F**. được gọi là dạng mệnh đề của **F**.

Định lý 9: INCONS(C) = INCONS(F)

Luật giải định giá cần thiết:

$$(\varphi, \alpha), (\psi, \beta) \vdash (R(\varphi, \psi), \min(\alpha, \beta)),$$

R là cái giải (Resolvent) của hai mệnh đề φ và α.

Định lý 10: C là tập các mệnh đề khả năng giá trị cần thiết, (c,α) là một mệnh đề khả năng nhận được bằng việc áp dụng một số hữu hạn luật giải (R) thì $C \models (c,\alpha)$.

Định lý 11 (Tính hợp lý và tính đầy đủ của phép loại trừ trong PL1): Giả sử **F** là một tập các công thức khả năng - tân từ cấp 1 giá trị cần thiết và C là dạng mệnh đề của **F**. Khi đó định giá của phép loại trừ tối ưu bởi phép giải từ C là mức độ mâu thuẫn của **F**.

Hệ quả: Giả sử φ là một công thức cổ điển, C là tập các mệnh đề nhận được từ $\mathbf{F} \cup \{(\neg\varphi,1)\}$. Định giá của phép loại trừ bởi phép giải tối ưu là Val(φ, **F**).

Loại trừ bởi phép giải:

1) Đưa F về dạng mệnh đề C.

2) Đưa φ về dạng mệnh đề, giả sử $c_1, c_2, \ldots, c_m$ là các mệnh đề nhận được.

3) Ký hiệu $C' = C \cup \{(c_1, 1), (c_2, 1), \ldots, (c_m, 1)\}$.

4) Áp dụng luật giải (R) trên C' để suy diễn ra (Θ, α)

5) Val(φ, **F**) = α.

Kết thúc phần trình bày này chúng tôi xin minh hoạ 1 ví dụ do D.Dubois và H.Prade [5] đề xuất:

Ví dụ: Xét một cơ sở tri thức khả năng giá trị cần thiết sau: $\mathbf{F} = \{\phi_i, i=1, \ldots, 6\}$ liên quan đến việc lựa chọn một ứng cử viên cho một nhiệm vụ nào đó.

ϕ_1: ((Elected(Peter) ∨ Elected(Mary)) ∧ (¬Elected(Peter) ∨ ¬Elected(Mary), 1).

ϕ_2: (∀ x ¬Current-president (x) ∨ Elected(x), 0.5),

ϕ_3: (Current-president (Mary, 1),

ϕ_4: (∀ x ¬Supports (John, x) ∨ Elected(x), 0.6),

ϕ_5: (Supports (John, Mary), 0.2),

ϕ_6: (∀ x ¬Victim-of-an-affair (x) ∨ Elected(x), ¬ 0.7),

F là tương đương với tập các mệnh đề khả năng $C = \{C_1, C_2, \ldots, C_7\}$.

C_1: (Elected(Peter) ∨ Elected(Mary), 1),

C_2: (¬Elected(Peter) ∨ ¬Elected(Mary), 1),

C_3: (¬Current-president (x) ∨ Elected(x), 0.5),

C_4: (Current-president (Mary, 1),

C_5: (¬Supports (John, x) ∨ Elected(x), 0.6),

C_6: (Supports (John, Mary), 0.2),

C_7: (¬Victim-of-an-affair (x) ∨ Elected(x), ¬ 0.7).

Chúng ta không tìm thấy bất kỳ sự loại trừ nào từ C, vì vậy C (cũng vậy **F**) là hoàn toàn phi mâu thuẫn. Giả sử ta muốn tìm biên dưới tốt nhất cho công thức ' Elected(Mary) ', ta ký hiệu

$C' = C \cup \{(C \text{ Elected(Mary)}, 1)\}$.

Lúc đó có thể áp dụng 2 luật giải cho $\{(\neg$ Elected(Mary). 1). C_3, $C_4\}$ và $\{(\neg$ Elected(Mary). 1), C_5, $C_6\}$ ta nhận được tương ứng (Θ, 0.5) và (Θ, 0.2), phép giải trước là tối ưu, phép giải sau là không tối ưu.

Chúng ta đang cần biết ai sẽ được lựa chọn và với mức độ cần thiết lớn nhất. Do

Val(Elected(Mary), F) = 0.5

điều đó chứng tỏ:

F |~~(Elected(Mary), 0.5);

Bằng cách tương tự ta cũng có:

F |~~($\neg$ Elected(Peter), 0.5);

Điều này cho ta kết luận ban đầu Mary có vẻ như có nhiều khả năng được lựa chọn hơn Peter. Bây giờ ta bổ sung thêm thông tin vào cơ sở tri thức đã cho:

ϕ_7 : (Victim-of-an-affair (Mary), 1.0).

Ký hiệu $F_1 = F \cup \{\phi_7\}$.

F_1 mâu thuẫn từng phần, và INCONS(FF$_1$) = 0.5. Ta cũng có thể chứng minh được rằng:

F_1 |= (Elected(Mary), 0.5);

F_1 |= ($\neg$ Elected(Peter), 0.5);

Nhưng những suy diễn này đã trở thành suy diễn tầm thường trên F_1. Trên F_1 thực ra ta có:

F_1 |= (Elected(Mary), 0.7);

F_1 |= ($\neg$ Elected(Peter), 0.7);

Do 0.7 > INCONS(F_1) ta có

F_1 |~~(Elected(Mary), 0.7);

F_1 |~~($\neg$ Elected(Peter), 0.7);

Điều đó có nghĩa là: bây giờ hầu như chắc chắn rằng Mary sẽ không được lựa chọn, trong khi Peter lại có thể. Như vậy việc cập nhật thêm thông tin đã dẫn đến một kết luận khá trái ngược với kết luận ban đầu.

3 Ngôn ngữ PL2 và một số vấn đề mở

3.1 Ngôn ngữ PL2 là ngôn ngữ bao gồm các công thức khả năng định giá cần thiết và công thức khả năng định giá khả năng.

$$F = \{ (S_i, a_i) / N(S_i) \geq a_i, a_i \in [0,1], i=1, \ldots, n \text{ và } (S'_i, b_i) / P(S'_k) \leq b_k, b_k \in [0,1], k=1, \ldots, m \}$$

Người ta đã đạt được một số kết quả ban đầu của ngôn ngữ này, như phân loại tính mâu thuẫn của F theo các mức: hoàn toàn mâu thuẫn, mâu thuẫn từng phần, mâu thuẫn yếu, hoàn toàn không mâu thuẫn dựa vào cả hai độ đo N và Π.

Và còn rất nhiều vấn đề cần được tiếp tục nhiên cứu mà trước hết là làm rõ ngữ nghĩa của ngôn ngữ này, liệu có thể xây dựng được phân bố khả năng đặc biệt như trong ngôn ngữ PL1 hay không, hệ thống hình thức của nó thể nào, ...

3.2 Ngôn ngữ PL1 với định giá không phải là số mà là ngôn ngữ.

3.3 Mối liên hệ giữa các lô gic: Khả năng, xác suất, modal lôgic, lý thuyết chứng cớ, ... Nhằm trả lời một số câu hỏi:

- Mỗi lý thuyết sẽ thích hợp trong những trường hợp nào? Một một hệ ứng dụng cần sử dụng nhiều mô hình biểu diễn thông tin suy luận khác nhau thì việc chuyển đổi biểu diễn và suy luận đó được thực hiện bằng cách nào ?, những rủi ro, phát sinh nào sẽ xẩy ra, ...
- Sử dụng các mối liên hệ đó để nghiên cứu lý thuyết này thông qua lý thuyết kia, ...

Tài liệu tham khảo

[1] **Boutilier C.**: Modal Logics for Qualitative Possibility Theory. Inter. Jour. of Approximate Reasoning. 1994. 10. 173-201.

[2] **Dubois D., H.Prade**: Posibility Theory: An approach to Computerized Procesing of Uncertainty, CNRS, LSI, Toulouse III (1986), bản dịch tiếng Anh, Univer. Of Cambridge. 1988.

[3] **Dubois D., H.Prade**: An Introduction to Possibilistic and Fuzzy Logics, Non standard Logics for Automated Reasoning (P.Smets and Al. Eds). Academic Press. New York. 1988. 287-326.

[4] **Dubois D. and H.Prade**: Epistemic Entrenchment and Possibilistic Logic, Artificial Intelligence, 1991, Vol 50. N1, 1-21.

[5] **Dubois D., J.Lang, H.Prade**: Possibilistic Logic. Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 3, Nonmonotonic reasoning and Uncertain Reasoning. Eds. Dov M.Gabbay, C.J.Hogger, J.A.Robinson, D.Nute. Clarendon Press. Oxford. 1994. 438-510.

[6] **NilsonN.J., Probabilistic Logic**: Artificial Intelligence, 1986, Vol 28, 21-87.

[7] **Thanh D.V.**: A relationship between the Possibilistic logic and the probabilistic logic. Computer and Artificial Intelligence. Vol. 17, No. 1, 51-68, 1998.

[8] **Zadeh L.A.**: Fuzzy Sets. Information and Control. 1965. 8, 338-353.

[9] **Zadeh L.A.**: Fuzzy Sets as basic for a theory of Possibility, Fuzzy Sets and Systems, 1978, Vol. 1, 3-28.

8 MỘT CÁCH TIẾP CẬN NGHIÊN CỨU PHÁT HIỆN TRI THỨC TRONG CÁC CƠ SỞ DỮ LIỆU TRỢ GIÚP QUYẾT ĐỊNH

Đỗ Văn Thành[a] - *Phạm Thọ Hoàn*[b]
[a]: *Văn phòng Chính phủ*
[b]: *Khoa Toán tin, Đại học Sư phạm Hà nội*

Tóm tắt

Mục đích của báo cáo này là trình bầy một số vấn đề nghiên cứu về hai giai đoạn quan trọng nhất của các quá trình từ cơ sở dữ liệu đến hỗ trợ ra quyết định một cách tự động là: Khai phá dữ liệu và Xây dựng các hệ thống thông minh tác nghiệp trên các dữ liệu được khai phá. Cách tiếp cận nghiên cứu của chúng tôi đối với hai giai đoạn này được cụ thể như sau: Xây dựng cơ sở tri thức từ cơ sở dữ liệu và Xây dựng cơ chế biểu diễn, lập luận trên cơ sở tri thức ấy. Trong bài báo sẽ trình bầy tóm tắt: Phương pháp khai phá dữ liệu, xây dựng cơ sở tri thức khả năng giá trị cần thiết từ một cơ sở dữ liệu quan hệ cho trước và đề xuất phương án giải quyết một bài toán mở trong lý thuyết khả năng khi định giá của độ đo khả năng không phải là số mà là giá trị ngôn ngữ.

1 Đặt vấn đề

Theo ý kiến của nhiều nhà nghiên cứu và ứng dụng tin học về cơ bản quá trình từ cơ sở dữ liệu (CSDL) đến ra quyết định có thể được xem là quá trình khai phá tri thức trong các CSDL (Knowledge Discovery Process in Databases: KDD). Quá trình này *nhằm kết xuất không tầm thường các thông tin tiềm ẩn, không biết trước, và tiềm năng có lợi từ các dữ liệu*. Nó là quá trình *tìm kiếm các mẫu tồn tại trong các CSDL, nhưng đang bị che dấu trong các khối dữ liệu*. Các mẫu này có thể cung cấp tri thức có giá trị, nếu dữ liệu đủ thuyết phục và chứa nhiều thuộc tính xác đáng đối với vấn đề đặt ra trong nhiều đại diện của phạm vi ứng dụng [1. 15. 27].

Về mặt công nghệ. hệ thống phát hiện tri thức là một môi trường được tích hợp, nó trợ giúp người dùng trong việc thực hiện quá trình phát hiện tri thức phức tạp từ khối dữ liệu đã cho [4, 15, 27].

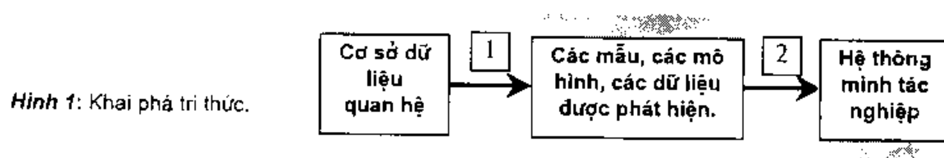
Quá trình KDD có thể được phân thành 5 giai đoạn: [1, 2. 15]

1) Xác định và định nghĩa vấn đề: Nhằm hiểu lĩnh vực ứng dụng, nhu cầu tri thức của người sử dụng, lựa chọn và tạo tập dữ liệu đích....
2) Nhận và tiền xử lý dữ liệu: Làm sạch dữ liệu, bỏ đi các dữ liệu tạp, các dữ liệu lỗi. không bình thường, xử lý dữ liệu bị mất, chuyển đổi dữ liệu sang dạng phù hợp, rút gọn

kích thước dữ liệu thông qua việc tìm các thuộc tính hữu ích, giảm bớt số chiều và biế- đổi dữ liệu để nhận được các bất biến....

3) Khai phá dữ liệu (Data-mining): Chọn nhiệm vụ khai phá dữ liệu, Lựa chọn các phương pháp khai phá dữ liệu. Thực hiện khai phá dữ liệu để rút ra các mẫu, các mô hình.
4) Giải thích kết quả và đánh giá các mẫu, các mô hình ở giai đoạn 3.
5) Sử dụng các tri thức được phát hiện: các tri thức được phát hiện được tích hợp trong một hệ thống, giải quyết các mẫu thuẫn tiềm tàng và nghiên cứu những thay đổi.

Một số thành viên của nhóm Xemina chúng tôi về "*Khai phá tri thức trong các Cơ sở dữ liệu*" của khoa Công nghệ, Đại học Quốc gia Hà nội tập trung nghiên cứu vào hai khâu trọng yếu của quá trình KDD như sau:



Hình 1: Khai phá tri thức.

Quá trình [1] từ cơ sở dữ liệu đến hình thành các mẫu biểu diễn dưới dạng luật thuộc về giai đoạn khai phá dữ liệu (Data mining) trong 5 giai đoạn trên. Trong khi nội dung chủ yếu của giai đoạn 4 và 5 được thể hiện trong quá trình [2] nhằm xây dựng các hệ thông minh tác nghiệp (Business Intelligence System) và đối với cách tiếp cận nghiên cứu của chúng tôi nội dung cơ bản của quá trình này là xây dựng cơ chế suy diễn trên các mẫu luật được phát hiện trong quá trình trước. Trong những nghiên cứu ban đầu chúng tôi chọn lý thuyết khả năng để biểu diễn các mẫu luật và để xây dựng cơ chế suy diễn trên các mẫu luật ấy. Các phần 2, 3 dưới đây sẽ trình bày khái quát cách tiếp cận nghiên cứu hai giai đoạn này.

2 Hình thành các mẫu luật trong lý thuyết khả năng từ cơ sở dữ liệu cho trước

2.1 Xuất xứ của vấn đề [5]: Xuất phát từ mong muốn giải quyết bài toán là: cho trước một CSDL quan hệ nào đó, người ta mong muốn xây dựng được các luật dạng: $\alpha\%$ của việc xảy ra của các sự kiện A_1, A_2, ... , và A_n, thì sẽ xảy ra sự kiện B, ở đây các A_i, B là tập các khoản mục (Itemset) được sinh từ CSDL quan hệ đã cho này. Phần mềm ứng dụng của nghiên cứu này được sử dụng để phát hiện và xây dựng chiến lược kinh doanh bán hàng, trong việc phân tích các hợp chất hữu cơ, nguyên tố vi lượng có trong các nguồn nước,....

Trong [5] người ta đã đề xuất cách giả quyết như sau: Giả sử $I = \{ i_1, i_2, \dots , i_m \}$ là tập của các chữ (literals) được gọi là các item. Giả sử D là tập của các tác vụ, mỗi một tác vụ T là một tập các item (ItemSet) sao cho $T \subseteq I$. Tác vụ T được gọi là chứa ItemSet X nếu $X \subseteq T$. Một luật kết hợp là một phép kéo theo dạng $X \Rightarrow Y$, ở đây $X \subseteq T$, $Y \subseteq T$ và $X \cap Y = \emptyset$.

Luật $X \Rightarrow Y$ xảy ra trong tập các tác vụ D với độ tin cậy (confidence) c nếu $c\%$ các tác vụ trong D chứa X thì nó cũng chứa Y. Luật $X \Rightarrow Y$ xảy ra trong tập các tác vụ D với độ hỗ trợ (support) s nếu $s\%$ các tác vụ trong D chứa $X \cup Y$. Bài toán được đề nghị trở thành:

Cho trước một tập các tác vụ, vấn đề phát hiện, khai phá tri thức ở đây là sinh ra tất cả các luật kết hợp với độ hỗ trợ và độ tin cậy cực tiểu (minSup, minConf), các số này được đưa ra bởi người dùng.

Thực tế trong ứng dụng: biểu diễn của D được quan tâm có thể là một tệp dữ liệu, cũng có thể là là một bảng dữ liệu quan hệ.

Vấn đề phát hiện ra các luật kết hợp được phân rã thành 2 vấn đề con:

a) Tìm tất cả các tổ hợp của các Item có các tác vụ hỗ trợ minSup ở trên. Người ta đã đề xuất 2 thuật toán Apiori và ApioriTid để giải quyết vấn đề này.

b) Sử dụng các ItemSet tìm được để sinh ra các luật mong muốn. Tư tưởng chính ở đây là ở chỗ: Nếu $ABCD$ và CD để là các ItemSet, nếu luật $AB \Rightarrow CD$ xảy ra bằng cách tính tỷ số r = support(ABC)/support(CD) thì chỉ khi có $r \geq minconf$ thì luật này mới được chấp nhận.

2.2 Đề nghị hình thành mẫu luật trong lý thuyết khả năng

Trước hết chúng tôi nhận xét rằng khái niệm độ tin cậy và độ hỗ trợ xuất phát từ ý tưởng của các khái niệm (được làm đơn giản hơn) Blief và Plausibility trong lý thuyết chứng cớ và niềm tin của Demster - Shafer [26]. Trong một số trường hợp đặc biệt các độ đo Blief và Plausibility có liên hệ mật thiết với nhau và trở thành độ đo xác suất, và trong trường hợp đặc biệt khác nó lại có thể trở thành độ đo cần thiết và độ đo khả năng trong lý thuyết khả năng.

Mặt khác: đặc trưng của các quá trình phát hiện tri thức là khối dữ liệu thường rất lớn, vì thế việc phát hiện tri thức thường được thực hiện trên các bộ phận của các khối dữ liệu đó và trong những trường hợp như vậy các luật được phát hiện thường không đầy đủ, không chắc chắn. Như đã biết hiện có 3 lý thuyết nổi trội nhất trong biểu diễn thông tin được biết không chắc chắn, không đầy đủ là lô gic xác xuất, lô gic mờ và lô gic khả năng. Trong nghiên cứu giải quyết bài toán ở trên, chúng tôi lựa chọn lô gic khả năng, mà cụ thể lô gic khả năng với giá trị cần thiết được xét trên ngôn ngữ cổ điển (ngôn ngữ mệnh đề, hoặc tân từ cấp 1 và được gọi là ngôn ngữ PL1) để biểu diễn các mẫu luật được phát hiện.

Nói cách khác, chúng tôi mong muốn sinh ra các luật dạng: Nếu xẩy ra các sự kiện A_1, A_2, ... , và A_n thì chắc chắn (khả năng) sẽ xẩy ra sự kiện B với độ tin cậy ít nhất ở mức α(nhiều nhất ở mức β). Về bản chất dạng luật này rất khác với dạng luật đã được nghiên cứu trong [5].

Chúng tôi đã đặt ra bài toán cần giải quyết sau đây: Cho trước một tập các tác vụ, vấn đề là sinh ra tất cả các luật kết hợp mức độ cần thiết cực đại và mức độ khả năng cực tiểu, các số này được đưa ra bởi người dùng.

Những nghiên cứu của chúng tôi mới là ban đầu và hiện chỉ tập trung vào việc tìm các luật kết hợp với một điều kiện là mức độ cần thiết cực tiểu.

2.3 Cách giải quyết và một số kết quả ban đầu

Để giải quyết bài toán đặt ra chúng tôi đã và đang giả 3 vấn đề sau đây:

a) Xây dựng độ đo cần thiết giá trị khoảng cho các ItemSet trong Cơ sở dữ liệu quan hệ cho trước.
b) Quan niệm mới về ngữ nghĩa của luật kết hợp
c) Trên cơ sở các thuật toán tìm các ItemSet và sinh luật đã biết, xây dựng các thuật toán đó khi các ItemSet và các luật kết hợp được do mức độ cần thiết.

Thực sự những nghiên cứu về những vấn đề này mới chỉ bắt đầu, chưa hoàn chỉnh, nhưng theo chúng tôi là có nhiều triển vọng tốt.

3 Lý thuyết khả năng mở rộng với định giá là giá trị ngôn ngữ

3.1 Những vấn đề mở trong lý thuyết khả năng:

Lý thuyết khả năng nguyên thuỷ được bắt nguồn từ lý thuyết mờ [7, 25] và được D.Dubois, H.Prade phát triển mạnh [7-11]. Nó mang nặng dấu ấn của lý thuyết tập mờ. Nhưng những phát triển về sau này của nó cho thấy lý thuyết này là một sự lai tạo giữa lý thuyết tập mờ và lý thuyết xác suất, nó vừa có mô hình toán học đẹp đẽ của lý thuyết xác suất, vừa có sự mềm dẻo của lý thuyết tập mờ [8-11]. Lý thuyết khả năng đặc biệt thích hợp để xử lý những thông tin chỉ được biết một phần, không được biết đầy đủ, hoặc mâu thuẫn từng phần trong khi lý thuyết xác suất rất thích hợp để biểu diễn và xử lý những thông tin được biết không chính xác, không chắc chắn và mang yếu tố ngẫu nhiên [6, 14]. Lý thuyết khả năng có 2 loại: định lượng và định tính. Lý thuyết khả năng định lượng xuất phát từ lý thuyết tập mờ, với ngữ nghĩa dựa trên các phân bố khả năng định lượng [25] (định giá là số thực trong [0, 1]) trong khi lý thuyết khả năng định tính xuất phát từ Modal logic có ngữ nghĩa dựa trên phân bố khả năng định tính là các quan hệ thự tự yếu [6]. Giữa hai loại lý thuyết khả năng này có nhiều mối quan hệ rất mật thiết với nhau. Trong trình bầy này chỉ đề cập đến lý thuyết khả năng định lượng.

Ký hiệu L={ S_i , i =1, ... ,n} là tập các mệnh đề. Khi đó F={(S_i , a_i)/ $N(S_i) \geq a_i$, $a_i \in$ [0,1], i =1, . .. n } (hoặc F*= {(S_i , b_i)/ $P(S_i) \leq b_i$, $b_i \in$ [0,1], i =1, ..., n }), ở đây N là một độ đo cần thiết (hoặc Π là độ đo khả năng), được gọi là cơ sở tri thức khả năng giá trị cần thiết (hoặc khả năng) giá trị khoảng. Ký hiệu **Ω** là tập tất cả các thế giới có thể đối với các mệnh đề trong **L** và nó cũng được gọi là tập các lớp thế giới có thể đối với các cơ sở tri thức khả năng **F**(**F***).

Mô hình lô gic khả năng:

Với mỗi cơ sở tri thức khả năng giá trị cần thiết **F**, nói chung có thể xác định được rất nhiều phân bố khả năng để thoả cơ sở tri thức đã cho, D.Dubois và H.Prade đã chỉ ra phương pháp để xác định phân bố khả năng ít đặc tả nhất thoả cơ sở tri thức này. Phân bố này là đặc trưng cho cơ sở tri thức đã cho và được xác định như sau:

$$\pi_{\mathbf{F}}(\omega) = \min \begin{cases} 1-\alpha & \text{nếu} \quad \omega \models \neg S_i \\ 1 \quad \text{nếu} & \omega \models S_1 \wedge S_2 \wedge \cdots \wedge S_n \end{cases}$$

với mọi $\omega \in \Omega$ [10-11], khi đó độ đo cần thiết $N_{\mathbf{F}}$ xác định từ phân bố này sẽ thoả: $N_{\mathbf{F}}(S_i)$ $\geq a_i$ với mọi i =1, ..., n.

Khi đó với của mệnh đề S bất kỳ, các độ đo khả năng và độ đo cần thiết của nó được xác định thông qua phân bố khả năng như sau:

$$\Pi(S) = \max_{\omega \in \Omega} \{ \pi_{\mathbf{F}}(\omega) / \omega \models S \} \qquad \text{và}$$

$$\Pi(S) = \min_{\omega \in \Omega} \{ 1-\pi_{\mathbf{F}}(\omega) / \omega \models \neg S \}.$$

Trong ngôn ngữ PL1 bao gồm các công thức khả năng định giá cần thiết, lý thuyết khả năng đã giải quyết khá trọn vẹn *những vấn đề cơ bản* sau:

1) Vấn đề suy diễn trong ngôn ngữ PL1.
2) Mâu thuẫn từng phần và suy diễn trong các cơ sở tri thức mâu thuẫn từng phần.
3) Hệ thống hình thức và vấn đề suy diễn tự động.
4) Tính hợp lý và tính đầy đủ mô hình lô gic khả năng ....

Trong [16-24] chúng tôi đã phát triển tiếp lý thuyết khả năng trong ngôn ngữ PL1, ở đó thực chất chúng tôi giả quyết bài toán sau:

Giả sử có m cơ sở tri thức khả năng giá trị cần thiết $\mathbf{F}_j$ ={(S_i , a_{ji})/ $N_j(S_i) \geq a_{ji}$, a_{ji} $\in$ [0,1] , j =1, ..., m, i =1, ..., n }, *khi đó với mệnh đề S bất kỳ ta đo mức độ khả năng và cần thiết của chúng bằng cách nào* ?. Việc giải quyết bài toán này cũng vẫn cần được tiếp

tục thêm. Chẳng hạn cần nghiên cứu đặc trưng của các toán tử tích hợp thoả các điều kiện đòi hỏi của các qúa trình tích hợp các phân bố khả năng ....

Trong lý thuyết khả năng D.Dubois and H.Prade đã nêu ra 2 bài toán mở sau đây:

a) **Ngôn ngữ PL2**: là ngôn ngữ bao gồm cả các công thức khả năng định giá cần thiết và lẫn công thức khả năng định giá khả năng:

$F=\{(S_i, a_i)/ N(S_i)\geq a_i, a_i\in[0,1], i=1,\ldots,n$ và $(S_i', b_i)/ P(S_k')\leq b_k, b_k\in[0,1], k=1,\ldots,m\}$.

Những vấn đề cơ bản đối với ngôn ngữ PL1 có giải quyết được trong ngôn ngữ PL2 không ?. Hiện nay khó khăn trước hết là ở chỗ chưa xây dựng được phân bố khả năng đặc trưng cho cơ sở tri thức hỗn hợp này.

b) **Trong ngôn ngữ PL1**: khi định giá không phải là số mà là giá trị ngôn ngữ, thì các vấn đề cơ bản trên sẽ xẩy ra như thế nào ? ...

Theo D.Dubois và H.Prade, giải pháp có lẽ là sử dụng modal logic và lô gic khả năng định tính để giải quyết vấn đề 2 [11], lý do là chưa có được một đại số ngôn ngữ đủ tốt có thể thay thế cho [0, 1] để làm định giá cho các độ đo thông tin. Một số tác giả đã đề xuất một cách tiếp cận khác để giải quyết vấn đề 2 thông qua việc lượng hoá các giá trị ngôn ngữ.

Cách tiếp cận của chúng tôi là khác với ý tưởng của D.Dubois và H.Prade và các tác giả đó.

c) **Ngoài ra những nghiên cứu về mối liên hệ giữa các lô gic**: Khả năng, Xác suất, Lý thuyết tập mờ, Modal Lôgic, Lý thuyết chứng cớ, ... cũng được đặc biệt quan tâm một mặt làm rõ ưu điểm và sức mạnh của từng loại lô gic trong những lớp bài toán cụ thể, mặt khác mong muốn sử dụng các mối liên hệ đó để nghiên cứu lý thuyết này thông qua lý thuyết kia. .... Kết quả đề án nghiên cứu mới đây của NASA đã chỉ ra rằng trên các tập vô hạn thì tính biểu diễn của 3 lô gic Khả năng, Xác suất, Mờ là tương đương, theo nghĩa thông tin được biểu diễn theo mô hình của lô gic này có thể biểu diễn được trong mô hình của lô gic kia với điều kiện ràng buộc nào đó và ngược lại. Nhưng trong trường hợp hữu hạn thì lô gic xác suất là mạnh hơn cả, đối lại lô gic xác suất phải trả giá bằng sự phức tạp tính toán của nó.

3.2 Ngôn ngữ PL1 với định giá là giá trị ngôn ngữ

Các cơ sở tri thức khả năng giá trị cần thiết được xét ở đây có dạng: $F=\{(S_i, x_i)/ N(S_i)\geq x_i, x_i\in X, i=1,\ldots,n\}$, ở đây $\underline{X}=(X, C, H, \leq, \neg, \cup, \cap, 0, 1)$ là đại số gia tử đối xứng hữu hạn (Symmetrical Hedge Algebra) được đề nghị bởi N.C. Ho và cộng sự [12-13]. Trong đó $C=\{F,W,T\}$, $F<W<T$ {(F,W,T) có thể là (false, unknow, true) hoặc (bad, normal, good) ...}.Hiện tại chúng tôi mới xét trường hợp các gia tử của đại số này H là hữu hạn và được

sắp tuyến tính. Tức là $H=\{h_1, h_2, \ldots, h_p, I; h_{p+1}, h_{p+2}, \ldots, h_{2p}\}$, và $h_1 < h_2 < \ldots < h_p < I < h_{p+1} < h_{p+2} < \ldots < h_{2p}$, $T=H_p[C]$. Ở đây $C \subset H[C] \subset H_2[C] \subset \ldots \subset H_p[C]=X$ và với mọi $x \in H_p[C]$ và $x \notin H_{p-1}[C]$ thì: $hx=x$ với mọi $h \in H$. X có phần tử lớn nhất là V^pTrue (Ký hiệu là 1) và phần tử nhỏ nhất là V^pFalse (Ký hiệu là 0).

Với mọi $h \in H$, $h(W)=W$; $\neg W=W$, $\neg T=F$, $\neg F=T$. Mọi $x \in T$, x đều có thể biểu diễn dưới dạng $x=h_1 h_2 \ldots h_i u$ $(u \in C)$ với $i \leq p$. Khi đó phủ định của x là $\neg x=h_1 h_2 \ldots h_i v$ với $v=\neg u$ trong C.

Mô hình lô gíc được đề nghị:

Với mỗi cơ sở tri thức khả năng giá trị cần thiết **F** như trên, chúng tôi đề xuất công thức xác định phân bố khả năng đặc biệt như sau:

$$\pi_F(\omega) = \min \begin{cases} \neg x_i & \text{nếu } \omega \models \neg S_i \\ 1 & \text{nếu } \omega \models S_1 \wedge S_2 \wedge \cdots \wedge S_n \end{cases}$$

với mọi $\omega \in \Omega$. Khi đó với của mệnh đề S bất kỳ, các độ đo khả năng và độ đo cần thiết của nó được xác định thông qua phân bố khả năng như sau:

$$\Pi(S) = \max_{\omega \in \Omega} \{\pi_F(\omega) / \omega \models S\} \quad \text{và}$$

$$\Pi(S) = \min_{\omega \in \Omega} \{\neg \pi_F(\omega) / \omega \models \neg S\}.$$

Khi đó độ đo cần thiết N_F xác định từ phân bố này sẽ thoả: $N_F(S_i) \geq x_i$, $\forall i=1, \ldots, n$.

Chúng tôi đã chứng minh được rằng phân bố được xây dựng theo công thức nêu trên là đặc tả ít nhất trong tất cả các phân bố khả năng nhận giá trị trong đại số gia tử đối xứng đặc biệt trên. Tức là nếu có phân bố khả năng π' thoả mãn điều kiện $N_F(S_i) \geq x_i$, $\forall i=1, \ldots, n$ thì $\pi' \geq \pi_F$.

Chúng tôi hy vọng là sử dụng phân bố này để nghiên cứu ngôn ngữ PL1 khi định giá là giá trị ngôn ngữ. hiện tại chúng tôi chưa xây dựng được hệ thống hình thức và các luật suy diễn, chưa chứng minh được tính hợp lý và tính đầy đủ của ngôn ngữ PL1, ….

Tài liệu tham khảo

[1] **Hồ Tú Bảo**: Giới thiệu về KDD và Datamining, Báo cáo Xemina: " Các vấn đề chọn lọc của Công nghệ Thông tin " giữa các Trường Đại học và Viện CNTT, 1997.

[2] **Đỗ Tấn Phong:** Một số nội dung nghiên cứu trong lập luận xấp xỉ. Luận văn Cao học Khoa Công nghệ thông tin, Đại học Khoa học Tự nhiên, Đại học Quốc gia Hà nội, 1999.

[3] **Đỗ Văn Thành:** Tích hợp ý kiên chuyên gia trong lý thuyết khả năng, Xemina: Những vấn đề chọn lọc về Công nghệ thông tin, Đại học Bách khoa Hà nội, 1998.

[4] **Đỗ Văn Thành:** Khai phá tri thức trong các các cơ sở dữ liệu, Báo cáo Xemina" Các hệ thống thông tin và hỗ trợ quyết định". Khoa Công nghệ, Đại học Quốc gia Hà nội, tháng 5/2000.

[5] **Agrawal R. et al.:** Fast Discovery of Association Rules, in Advances in Knowledge Discovery and Data mining, AAAI Press/ The MIT Press, USA, 1996.

[6] **Boutilier C.:** Modal Logics for Qualitative Possibility Theory, Inter. Jour. of Approximate Reasoning, 1994, 10, 173-201.

[7] **Dubois D., H.Prade:** Posibility Theory: An approach to Computerized Procesing of Uncertainty. CNRS, LSI, Toulouse III (1986), bản dịch tiếng Anh, Univer. Of Cambridge, 1988.

[8] **Dubois D., H.Prade:** An Introduction to Possibilistic and Fuzzy Logics, Non standard Logics for Automated Reasoning (P.Smets and Al. Eds). Academic Press, New York, 1988, 287-326.

[9] **Dubois D. and H.Prade.:** Aggregation of possibility measures, Multiperson decision Making using fuzzy sets and Possibility Theory, (J.kacprzyk and M.Fedrizzi, Eds.) Kluwer Academic Publ., 1990, 55-63.

[10] **Dubois D., H.Prade:**Necessity measures and the Resolution Principle. IEEE Tras. on Systems, Man and Cybernetics, 1987, 17, 474-478.

[11] **Dubois D., J.Lang, H.Prade:** Possibilistic Logic, Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 3, Nonmonotonic reasoning and Uncertain Reasoning. Eds. Dov M.Gabbay, C.J.Hogger, J.A.Robinson, D.Nute. Clarendon Press, Oxford, 1994, 438-510.

[12] **Ho. N.C, Nam. H.V, Chau. N.H:** Hedge Algebra, linguitic-valued logic and their Application to Fuzzy Reasoning. Inter. Jour. of Uncertainty., Fuzzyness and Knowledge Based Systems. Vol. 7. N 4 (1999), 347-361.

[13] **Ho. N.C, Nam. H.V:** A Theory of refinement Structure of Hedge Algebra and its Application to Fuzzy Logic. Logic. Algebra and Computer Science. Banach Center Publication, Vol. 46, 1999, 63-90.

[14] **Nilson N.J.:** Probabilistic Logic, Artificial Intelligence, 1986, Vol 28, 21-87.

[15] **Roanal J., Brachman and al.:** Knowledge Discovery Process in Databases. Advances in Knowledge Discovery and Data mining, AAAI Press/ The MIT Press, USA, 1996.

[16] **Thanh D.V.:** Reasoning Methods on Multiperson Knowledge Bases. Information Technology: Reseach and Application, 1996, 403-419, 20th Anniversary Memorial Volume of Institute for IT(In Vietnamese).

[17] **Thanh D.V.:** Aggregation of distributions and Aggregation operators. Computer and Cybernetics, Tom 12, 3 (1996), 47-63.

[18] **Thanh D.V.:** Hierarchical Aggregation of Possibility Distributions. Proceedings of the National Center for Science and Technology., Vol. 9, N1, 1997, PP 29-41.

[19] **Thanh, D.V.:** Stability of the principles of minimal Specificity and maximal Buoancy: presentation in The joint Pacific Asian Conference on Expert System/ Singapore International Conference on Intelligent Systems, Singapore, 24-27 February 1997.

[20] **Thanh D.V.:** A relationship between the Possibilistic logic and the probabilistic logic. Computer and Artificial Intelligence, Vol. 17, No. 1, 51-68, 1998.

[21] **Thanh.D.V.:** Possibility Consensus Model, Proceedings of Japan -Vietnam Bilateral Symposium on Fuzzy Systems and Appliacations, 30th September - 2th October 1998, pp 288-293.

[22] **Thanh. D.V.:** Possibility Information Measures and Selection Approach, Computer and Artificial Intelligence, Vol. 18, 1999, N6, 595-610.

[23] **Thanh. D.V.:** Aggregation of Expert Knowledge for Medical Diagnosis. Proceeding of the 8th International Fuzzy Systems Association World Congress, IFSA'99, Taiwan, Vol. 1, 279-283.

[24] **Thanh D.V.:** Aggregation of Possibility distributions proceeded via Probabilistic Models Logic. (to appear in Advances of Natural Sciences, N2, 2000).

[25] **Zadeh L.A.:** Fuzzy Sets as basic for a theory of Possibility, Fuzzy Sets and Systems, 1978, Vol. 1, 3-28.

[26] **Wagner C. G.**: Consensus for Belief Functions and related uncertainty measures. Report ORNL/TM-10748, Oak Ride, Tenn., 1988

[27] **Willi Klosgen. Explora:** A Multipattern and Multistrategy Discovery Assistant. In Advances in Knowledge Discovery and Data mining. AAAI Press/ The MIT Press, USA, 1996. 248-271.

9 TÍCH HỢP CÁC KỸ THUẬT TÍNH TOÁN MỀM VÀ MẠNG NƠRON TRONG XỬ LÝ DỮ LIỆU

Nguyễn Thanh Thuỷ và Nguyễn Hữu Đức
Khoa Công nghệ Thông tin - Đại học Bách Khoa Hà Nội
Trần Ngọc Hà
Viện Công nghệ Xạ hiếm

1 Xử lý dữ liệu trong các ứng dụng tin học

Trong các ứng dụng tin học, nhu cầu xử lý dữ liệu bao gồm quá trình phân tích dữ liệu mẫu , mô hình hoá dữ liệu hay khai thác xử dụng những kho dữ liệu lớn là một nhu cầu cấp thiết cho những hệ thống ứng dụng đòi hỏi tính mềm dẻo. Ta hãy thử phân tích một vài ứng dụng để thấy rõ được nhu cầu cấp thiết này.

Trong lĩnh vực tài chính thương mại, do độ rủi ro cao của quá trình đầu tư cho các dự án, người ta thường nghĩ đến việc xây dựng các ứng dụng cho phép các nhà đầu tư có thể đánh giá được các điều kiện cho môi trường đầu tư. Việc đánh giá này đòi hỏi ứng dụng không những phải khai thác những yếu tố khách quan như khả năng tiêu thụ của khách hàng, khả năng cung cấp của các nhà cung ứng vật tư, thông số về tình hình thị trường chứng khoán... mà còn phải khai thác cả những yếu tố kinh nghiệm của các dự án đã thực hiện trong cùng môi trường.

Một thí dụ khác cũng được đưa ra phân tích là ứng dụng dự báo thời tiết cho nghành khí tượng thuỷ văn. Công việc dự báo thời tiết đòi hỏi phải tập trung rất nhiều thông tin thời tiết từ các trạm đo ở khắp cả nước. Thông thường người ta áp dụng những phương pháp lý thuyết đã được công nhận để phân tích và đưa ra dự báo cuối cùng. Tuy nhiên do sự thay đổi liên tục về quy luật khí hậu cũng như việc thu thập thông tin không đầy đủ dẫn đến độ chính xác của các phương pháp này thường không cao. Một đề xuất đặt ra cho công việc dự báo này là xây dựng một ứng dụng có khả năng học được những quy luật thay đổi thời tiết dựa trên những tình huống đã xảy ra trong quá khứ. Để thực hiện điều này, ứng dụng cũng phải khai thác một khối lượng khổng lồ dữ liệu quá khứ với những cấu trúc phức tạp, không đồng nhất và đôi khi không chính xác và thiếu đầy đủ.

Trong thực tế, có vô vàn những ứng dụng có cùng nhu cầu xử lý dữ liệu như vậy. Chìa khoá cho những thành công của các ứng dụng này chính là những phương pháp tiên tiến để khai thác dữ liệu một cách hiệu quả. Chính nhu cầu này là động cơ chính thúc đẩy cho sự hình thành và phát triển của một chuyên ngành mới – chuyên ngành khai khoáng dữ liệu (data mining). Trong một kho dữ liệu khổng lồ, người ta tìm cách trích chọn, đúc rút ra

được những thông tin quan trọng và cần thiết cho nhu cầu của ứng dụng. Khi những thông tin này mang một tính quy luật và có thể coi là những *tri thức*, những kinh nghiệm quý báu được tổng kết thì quá trình khai khoáng dữ liệu trở thành một quá trình phát hiện tri thức từ dữ liệu (Knowledge Discovering in Data).

Trong lĩnh vực này (khai khoáng dữ liệu và phát hiện tri thức từ dữ liệu), rất nhiều phương pháp, kỹ thuật đã được áp dụng nhằm tìm ra những quy luật, những mẫu hình ẩn dấu bên trong một khối dữ liệu hỗn độn. Có thể liệt kê ra ở đây một vài phương pháp trong rất nhiều các phương pháp được áp dụng trong lĩnh vực này:

- Tìm kiếm quan hệ giữa các thuộc tính bằng phương pháp tương quan và hồi quy của thống kê.
- Biểu diễn tri thức dưới dạng luật kết hợp.
- Xắp xếp phân loại tri thức nhờ cây quyết định.
- Biểu diễn tri thức dưới mô hình mạng Bayes.

...

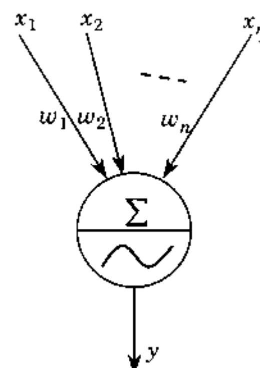
Và ở đây cũng cần đề cập tới một công cụ quan trọng và hiệu quả cho quá trình khai thác dữ liệu – mô hình *mạng nơron*.

2 Tiếp cận mạng nơron trong xử lý dữ liệu

Nói đến mạng nơron ở đây ta hiểu là mạng nơron nhân tạo (artificiel neuron network) – sự mô phỏng toán học của các mạng nơron sinh học (biological neuron network). Một mạng nơron nhân tạo được xây dựng từ những thành phần cơ sở là những nơron nhân tạo được mô tả như hình 1.

Mỗi nơron nhân tạo này có nhiều đầu vào và một đầu ra. Các đầu vào tiếp nhận kích thích từ đầu ra của những nơron khác hoặc từ môi trường. Mỗi nơron vào có một trọng số (weight) nhằm khuyếch đại tín hiệu kích thích. Tất cả các tín hiệu kích thích này sau khi được khuếch đại sẽ đi vào thân nơron bắt đầu từ bộ cộng. Bộ cộng sẽ thực hiện việc tổng hợp các kích thích theo công thức :

$$Net = \sum_{i=1}^{n} x_i w_i \, .$$



Hình 1: Nơron nhân tạo

Tín hiệu sau khi tổng hợp được lại tiếp tục được biến đổi nhờ một hàm phi tuyến, thường được gọi là hàm kích hoạt (activation function).

Dưới đây là một số hàm kích hoạt thường được sử dụng :

– Hard-limit $hl(Net,\theta) = \begin{cases} 0 & \text{nếu} \quad Net \geq \theta \\ 1 & \text{nếu} \quad Net < \theta \end{cases}$.

– Thresold $tl(Net,\alpha,\theta) = \max(1,\alpha(Net+\theta))$.

– Sigmoidal $sf(Net,\alpha,\theta) = \dfrac{1}{1+e^{-\alpha(Net+\theta)}}$.

Và cuối cùng tín hiệu sẽ được đưa ra đầu ra của mạng nơron để lại trở thành đầu vào của các nơron khác hoặc trở thành tín hiệu ra của toàn bộ mạng.

Khi kết hợp các nơron lại với nhau thành một mạng nơron nhân tạo, ta có thể xem như mạng này là một mô hình tính toán $Y=F(X)$ với X là véc tơ số liệu vào và Y là véc tơ số liệu ra. Ưu điểm của một mạng nơron nhân tạo như vậy là nó cho phép xây dựng một mô hình tính toán có khả năng học dữ liệu rất cao. Chỉ cần đưa vào cho mạng một tập mẫu dữ liệu trong quá trình học là mạng có khả năng phát hiện những ràng buộc dữ liệu và áp dụng những ràng buộc này trong quá trình xử dụng mà không cần phải có thêm các tri thức về miền ứng dụng. Khả năng này cho phép ta xây dựng mô hình dữ liệu một cách dễ dàng. Một ưu điểm khác của mạng nơron là khả năng dung thứ lỗi cao. Mạng có thể chấp nhận những dữ liệu mẫu không hoàn toàn chính xác tuyệt đối mà vẫn đảm bảo được phần nhiều tính đúng đắn của bài toán. Điều này làm giảm nhẹ rất nhiều quá trình sàng lọc, làm mịn dữ liệu trong khai thác. Với những đặc điểm này, mạng nơron cho phép dễ dàng xây dựng các mô hình thích nghi mà trong đó sự thay đổi liên tục về quy luật dữ liệu có thể dễ dàng được ập nhật trong quá trình học lại của mạng. Tuy nhiên mạng nơron cũng không phải là một công cụ vạn năng. Nó cũng có một số nhược điểm khó khắc phục:

– Mạng chỉ có thể làm việc với những dữ liệu số.

– Để mạng đạt được một hiệu quả cao thì điều cần thiết là phải có một bộ dữ liệu mẫu đủ lớn cho quá trình học.

– Mạng chỉ có tính chất nội suy. Khả năng ngoại suy là rất kém.

– Mạng không thể đưa ra được cơ chế giải thích.

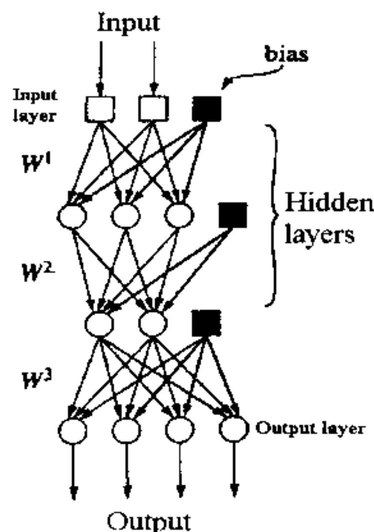
– Các giải thuật học của mạng đôi khi chưa đảm bảo sự hội tụ cần thiết cho quá trình sử dụng.

Những nhược điểm này chính là một trong những tiền đề cho ý tưởng tích hợp mạng nơron với các kỹ thuật khác của chúng tôi. Trong phạm vi của bài báo chúng tôi xin chỉ đề cập tới những nghiên cứu của nhóm về mạng nơron nhiều lớp lan truyền thẳng (multi-layer feedforward nơ-ron network) với giải thuật học lan truyền ngược lỗi (back-propagation of error - BP) và những cải tiến nhằm khắc phục nhược điểm của mạng này.

3 Mạng nơron nhiều lớp truyền thẳng và giải thuật học BP

3.1 Kiến trúc

Một mạng nơron nhiều lớp lan truyền thẳng có thể được mô tả như hình 2. Mạng này có đặc tính là chia thành nhiều lớp , lớp vào tương ứng với các tín hiệu đầu vào của hệ thống và lớp ra tương ứng với các kết quả trả về. Các nơron giữa các lớp kế tiếp nhau được nối một các đầy đủ. Mặc dù một mạng nơron tổng quát, người ta có thể sử dụng nhiều lớp ẩn tuy nhiên người ta đã chứng minh được rằng chỉ cần một lớp ẩn của mạng là đủ mô hình hoá một hàm bất kỳ nào đó [3].



Hình 2: Mạng nhiều lớp truyền thẳng

Mạng nơron này cũng có thể được đặc tả bởi các tập các thông tin cấu trúc S bao gồm số lớp của mạng, số lượng nơron trên một lớp và một tập thông tin tham số P chính là tập các trọng số của từng nơron trong mạng. Khi đó để xây dựng một mạng ta cần phải xác định cả hai tập S và P này. Thông thường , qua trình xác định S và P phụ thuộc vào tập những dữ liệu mẫu (X_i, Y_i) đã được lưu trữ lại từ những tình huống sử dụng thực tế của hệ thống. Mạng sẽ được học nhằm xây dựng được một cặp S và P tốt nhất và từ đó nó có khả năng dự đoán một kết quả Y khi có một đầu vào X nào đó sao cho bộ (X, Y) này cũng đảm bảo được tính quy luật như trong tập dữ liệu mẫu. Như đã đề cập ở trên, thông thường một mạng nơron chỉ có một lớp ẩn, khi đó công việc xác định bộ thông số S chỉ là việc chỉ ra cụ thể xem lớp ẩn có bao nhiêu nơron. Việc xác định này thường được thực hiện dựa trên kinh nghiệm hoặc phép thử và sai và thực ra ta cũng không có quá nhiều lựa chọn cho tham số S này. Chính vì vậy ta tập trung ở đây vào công việc xác định bộ tham số P. Một phương pháp xác định truyền thống nhất đó là áp dụng giải thuật học BP.

3.2 Giải thuật học lan truyền ngược lỗi (Back-propagation of error)

Giải thuật học này được xem như một giải thuật học có thầy (suppervised learning) nghĩa là để mạng có thể học được thì bộ dữ liệu mẫu phải là tập những cặp (X_i, Y_i). Với mỗi cặp dữ liệu mẫu (X_i, Y_i) này giải thuật sẽ thực hiện hai pha:

Pha1 : X_i được đưa vào mạng và được la truyền trong mạng để thu được một đầu ra O_i.

Pha2 : Hàm lỗi $E_i = (Y_i - O_i)^2$ được tính toán và lỗi này lại được lan truyền ngược lại đầu ra của mạng nhằm mục đích cập nhật lại các trọng số của mạng.

Để thu được một kết quả tốt, mạng sẽ phải thực hiện hàng ngàn vòng lặp như vậy để cuối cùng thu được một bộ trọng số sao cho sai số E là nhỏ nhất.

3.3 Gọi lại (recall) và dự báo (prediction)

Gọi lại là khả năng dự đoán đầu ra của mạng trong quá trình học còn dự báo là khả năng dự đoán của mạng trong quá trình xử dụng. Để xác định được các khả năng này người ta thường chia bộ mẫu học thành 2 phần T_{train} và T_{test}. Chỉ có T_{train} được sử dụng cho quá trình học thôi còn T_{test} được sử dụng để thử dự báo. Kết quả được đánh giá bởi hàm sai số bình phương tối thiểu RMS :

$$RMS = \sqrt{\frac{\sum_{s=1}^{p}\sum_{i=1}^{n}(y_{si} - O_{si})^2}{pn}}$$ trong đó p là số lượng mẫu và n là số đầu ra của mạng

Hàm sai số này sẽ được tính toán trên cả hai tập T_{train} và T_{test} và sẽ đại diện cho khả năng gọi lại và dự báo của hệ thống. Thông thường nếu kết quả học là tốt thì cả hai hàm sai số này đều cho những giá trị sai số rất nhỏ. tuy nhiên cũng có khi sai số học thì nhỏ mà sai số khi dự báo thì lớn (overtraining effect). Điều này cho phép ta khẳng định là có sự dư thừa trong các tham số cấu trúc mà cụ thể là số lượng nơron lớp ẩn quá nhiều. Khi đó ta phải thay đổi lại số lượng nơron này và bắt đầu lại một quá trình học khác.

4 Quan điểm toán học về quá trình học của mạng nơron

Theo chúng tôi , một quá trình học của mạng nơron bao thực chất là một quá trình tìm kiếm một bộ các tham số trên một không gian của các tham số đó. Quá trình này cần phải thực hiện theo hai bước :

- Xác định hàm giá trên các tham số
- Tối thiểu hoá hàm giá trong không gian của các tham số

4.1 Học tham số

Học tham số là quá trình xác định một tập tham số P (hay là các trọng số) tốt nhất với một cấu trúc mạng cố định. Để làm được điều này, ta cần phải xây dựng một hàm giá dựa trên tập các dữ liệu mẫu T_{train} và tập trọng số W. Hàm giá này có thể là một hàm khả vi bất kỳ mà có tính chất đạt đến cực tiểu khi các đầu ra O_i đúng bằng đầu ra lý tưởng Y_i của tập mẫu. Ta có thể xây dựng hàm giá này dưới dạng L_n – norm như sau:

$$W = \frac{1}{p}\sum_i (y_i - O_i)^p \quad \text{với} \quad 1 \le p \le \infty.$$

Với hàm giá này, ta có thể áp dụng một giải thuật tìm kiếm nào đó trên không gian R^m của tập trọng số. Nếu thu được kết quả tốt với một cực tiểu toàn cục, ta sẽ có một bộ tham số tốt nhất cho mạng.

4.2 Học tham số bằng giải thuật lan truyền ngược lỗi (BP)

Trong giải thuật BP, với mỗi một cặp dữ liệu mẫu (X_i, Y_i) người ta xây dựng một hàm giá như sau:

$$E(w) = \frac{1}{2}\sum_{j=1}^{n}(y_j - O_j)^2 \ .$$

Nếu như ta có p cặp dữ liệu mẫu như vậy, ta sẽ có p hàm giá khác nhau và các hàm giá này sẽ được tối thiểu hoá từng bước trong quá trình học của mạng bằng phương pháp gradient:

$$\Delta w_{ji}^{l} = -\eta \frac{\partial E}{\partial w_{ji}^{l}}$$

$$\Delta w_{ji}^{l}(n) = \eta \delta_j(n) y_i^{\ l}(n)$$

trong đó Δw_{ji}^{l} là sự điều chỉnh trọng số sau mỗi bước lặp l của nơron thứ j trong lớp i. η gọi là hệ số học hay tốc độ học.

Tuy nhiên chúng ta cũng được biết rằng $E(W)$ là một hàm có bề mặt rất phức tạp. Nếu như với một mạng một lớp (không có lớp ẩn) thì bề mặt này có hình dạng paraboloid và khi đó phương pháp gradient mới chắc chắn cho ta đạt tới một cực trị toàn cục của hàm giá. Tuy nhiên nếu số lượng lớp ẩn lớn hơn hoặc bằng một thì bề mặt này trở thành một bề mặt xoắn có nghĩa là nó tồn tại vô số cực trị địa phương và do đó phương pháp gradient không thể đảm bảo cho ta tìm thấy được một cực trị toàn cục trên bề mặt hàm giá này. Chính vì vậy để cải thiện giải thuật gradient này, người ta thường tìm cách thay đổi hệ số học hoặc thêm vào đó thành phần quá tính để cho phép có thể vượt qua những cực trị địa phương trong quá trình tìm kiếm.

$$\Delta w_{ji}^{l}(n) = \eta \delta_j(n) y_i^{\ l}(n)(3) + \alpha \Delta w_{ji}^{l}(n-1) \ .$$

Việc lựa chọn hệ số học η hoặc hệ số quán tính α cũng là một vấn đề rất khó khăn bởi vì nếu hệ số học quá lớn đôi khi dẫn đến tình trạng không ổn định của quá trình tìm kiếm. Còn hệ số học quá nhỏ lại dẫn đến tốc độ học chậm và khả năng vượt qua các cực trị địa phương thấp. Quá trình tìm kiếm trên những hàm giá như thế này đã được chứng minh là một bài toán NP-đầy đủ nghĩa là ta không thể xử dụng được một giải pháp tổng quát có độ phức tạp đa thức để đạt đến kết quả. Xuất phát từ quan điểm này, trong bài báo chúng tôi đề xuất việc tính hợp các kỹ thuật tìm kiếm toàn cục dựa trên một số yếu tố ngẫu nhiên

(stochaostic) của SoftComputing với giải thuật học của mạng nơron truyền thẳng nhiều lớp để có thể đạt tới được một kết quả tốt nhất có thể.

5 Tích hợp giải thuật di truyền với quá trình học của mạng nơron nhiều lớp truyền thẳng

Giải thuật di truyền được biết đến như một giải thuật tìm kiếm dựa trên học thuyết về chọn lọc tự nhiên và nó cho phép ta đạt được tới những cực trị toàn cục [1] . Để áp dụng được giải thuật di truyền trong quá trình tìm kiếm trên không gian tham số R^m. Ta cần phải thực hiện các bước sau :

1) *Mã hoá các trọng số của mạng nơron thành các nhiễm sắc thể đại diện cho các cá thể của một quần thể của giải thuật di truyền.*

Mỗi cá thể trong giải thuật di truyền sẽ thay mặt cho một bộ trọng số của mạng nơron. Mỗi trọng số này sẽ được mã hoá thành một đoạn k-gen trong nhiễm sắc thể tương ứng. Ở đây ta không cần phải phân biệt trọng số nào ở lớp nào mà ta chỉ cần trải tất cả các trọng số lên sơ đồ gen của nhiễm sắc thể. Do tiêu chuẩn về minimal alphabets [1] nên mỗi nhiễm sắc thể sẽ được mã hoá thành một chuỗi cố định các gen và do đó các trọng số cần phải được nắn (scale) cho phù hợp với khung của từng đoạn gen. Ví dụ một nhiễm sắc thể có độ dài 20 bit biểu diễn 5 trọng số thì mỗi trọng số cần phải được nắn chỉnh về 4 bit tức là về giá trị trong khoảng 0 đến 15.

Sau khi đã thu được kết quả, các trọng số sẽ được chuyển đổi ngược lại về giá trị thực qua quá trình giải mã.

2) *Xây dựng hàm giá*

Hàm giá này sẽ được sử dụng để tạo nên độ phù hợp (fitness) của các cá thể và của cả quần thể trong giải thuật di truyền.

Khác với giải thuật BP (sử dụng nhiều hàm giá), giải thuật di truyền sử dụng duy nhất một hàm giá. Nếu như ta có T_{train} là tập các dữ liệu mẫu thì hàm giá sẽ được tính toán trên toàn bộ tập này như sau :

$$E(w)=\frac{1}{2}\sum_{s=1}^{P}\sum_{j=1}^{n}(y_{sj}-O_{sj})^2$$

Trong quá trình tiến hoá của toàn bộ quần thể, hàm giá này sẽ dần dần đạt tới cực tiểu toàn cục.

3) *Khởi tạo quần thể đầu tiên*

Theo C.T. Lin, C.S.G. Lee [2] quần thể đầu tiên sẽ đạt hiệu quả tốt nhất khi ta lựa chọn ngẫu nhiên với phân bố xác xuất $e^{-|x|}$ trong đó x là số lượng các giải pháp chọn. Đây là

một quan sát thực tiễn cho phép hạn chế được không gian lựa chọn ban đầu của các cá thể trong quần thể.

4) Kết hợp giải thuật di truyền với BP

Mặc dù giải thuật di truyền có khả năng đạt được tới cực trị toàn cục cho quá trình tìm kiếm nhưng do có kết hợp những yếu tố ngẫu nhiên nên tốc độ tìm kiếm nói chung là rất chậm. Mặt khác nó không thể hoàn toàn đạt được tới cực trị toàn cục mà chỉ cho những kết quả *xung quanh đó*. Đối lập với giải thuật di truyền, BP lại cho phép đạt được những cực trị nếu như điểm xuất phát của quá trình tìm kiếm nằm trong vùng của cực trị toàn cục. Từ đó chúng tôi có ý tưởng kết hợp cả giải thuật di truyền và BP nhằm đạt tới một kết quả trọn vẹn trong bài toán tìm kiếm.



Hình 3 :Kết hợp GA và BP

Trong giải thuật kết hợp này, giải thuật di truyền được sử dụng như một bộ khởi tạo cho BP. Tập trong số được mã hoá thành các nhiễm sắc thể và được tiến hoá nhờ giải thuật di truyền. Kết thúc quá trình tiến hoá , bộ trọng số tốt nhất tương ứng với cá thể ưu việt nhất trong quần thể được lựa chọn làm những trọng số khởi tạo cho giải thuật BP. Nó chính là bộ tham số cho phép xác định điểm gần cực trị toàn cục nhất của hàm giá.

Với sự kết hợp này, giải thuật BP sẽ cần phải được thay đổi một vài yếu tố :

- Giải thuật không tự khởi tạo trọng số mà nhận các trọng số từ giải thuật di truyền
- Thành phần quán tính được loại bỏ để làm tăng tốc độ của quá trình hội tụ và loại bỏ dao động.
- Hệ số học thích nghi [1] được áp dụng để tăng tốc độ tìm kiếm

$$\Delta\eta = \begin{cases} +a & \text{nếu } \Delta E < 0 \text{ duy trì trong } k \text{ bước} \\ -b\eta & \text{nếu } \Delta E > 0 \\ 0 & \text{các trường hợp khác} \end{cases}$$

Trong đó $\Delta\eta$ là sự điều chỉnh hệ số học sau mỗi bước tính toán.

5) Những nhược điểm của giải thuật di truyền

Sở dĩ giải thuật di truyền có một tốc độ hội tụ chậm là do sự kém hiệu quả của các toán tử lai ghép (crossover) và biến dị(mutation). Trong các toán tử này, các nhiễm sắc thể được tạo ra đôi khi rất "yếu". Do vậy quá trình tiến hoá của toàn bộ quần thể sẽ bị chậm lại. Nguyên nhân chính là do hai toán tử này xây dựng các nhiễm sắc thể mới dựa trên yếu tố ngẫu nhiên mà không cần biết chúng có hiệu quả hay không.

Chính vì vậy cần phải tìm cách cải thiện hai toán tử này mà không làm mất đi yếu tố ngẫu nhiên trong đó. Giải thuật mô phỏng quá trình tôi thép (Simulated Annealing – SA) là lựa chọn của chúng tôi nhằm cải tiến hai toán tử này.

6 Cải thiện giải thuật di truyền bằng mô phỏng quá trình tôi thép

Khác với giải thuật di truyền, mô phỏng quá trình tôi thép không sử dụng phương pháp tìm kiếm trên một quần thể mà chỉ xuất phát từ một điểm giống như các phương pháp gradient. Tuy nhiên thông qua việc chuyển trạng thái bằng lịch trình tôi và xác xuất chuyển trạng thái thì yếu tố ngẫu nhiên được đưa vào cho phép giải thuật từ xuất phát điểm ban đầu có thể vượt qua những vùng cực trị cục bộ để đạt tới cực trị toàn cục. Một điểm khác nữa giữa SA và GA đó là sự xác định tương đối trong quá trình tìm kiếm. Với GA ta không thể xác định được với baonhiêu thế hệ thì có thể đạt tới trạng thái ưu việt tuy nhiên trong SA quá trình tiến đến trạng thái ưu việt hoàn toàn được xác định thông qua lịch trình tôi. Chính vì vậy chúng tôi muốn được kết hợp hai phương pháp này để chúng có thể bổ xung cho nhau những ưu điểm của từng phương pháp.

Cụ thể của việc tích hợp là sự thay thế của hai toán tử Crossover và Mutation bằng hai toán tử mới SA-Crossover và SA-Mutation.

```
Procedure SAC(parent1,parent2,T) {
      crossover( parent1,parent2->child1,child2)
      if C_accept(child1,child2,parent1,parent2,T) then return child1,child2
      else return parent1,parent2 }
Procedure SAM(ch,T) {
      new_ch=mutate(ch,T)
      if  M_accept(ch,new_ch,T) then return new_ch
      else return ch }
```

trong đó crossover và mutation là hai toán tử truyền thống của GA. Thay vì sử dụng ngay kết quả của hai toán tử này thì ta cần lọc qua bằng các hàm *C_accept* và *M_accept*. Đây chính là điểm kết hợp giữa SA và GA.

Nếu như E và new_E là những hàm giá của các nhiễm sắc thể cũ và nhiễm sắc thể mới tạo ra bởi các hàm truyền thống thì

$$\Delta E = new_E - E$$

chỉ ra sự chênh lệch về độ phù hợp giữa hai nhiễm sắc thể này.

SA sẽ sử dụng ΔE để quyết định có sử dụng nhiễm sắc thể mới hay không thông qua phân bố xác xuất Metropolis

$$\begin{matrix} \text{M_accept} \\ = \\ \text{C_accept} \end{matrix} \begin{cases} true & if\ \Delta E \le 0 \\ true & if\ \Delta E > 0\ and\ z \le p = \exp(-\Delta E / T) \\ false & otherwise \end{cases}$$

z được sử dụng ở đây là một số ngẫu nhiên trong khoảng 0 – 1 để quyết định sự chuyển trạng thái bằng phân bố xác xuất.

Ta có thể thấy rõ ràng: Nếu nhiệt độ T cao thì khả năng chuyển trạng thái là có thể xảy ra dễ dàng nhưng nếu T thấp thì khả năng chuyển trạng thái là rất thấp nếu như các toán tử truyền thống cho một kết quả tồi hơn.

Lịch trình tôi cũng được xác định thông qua công thức do Geman S. and D. Geman [1984] đưa ra

$T(g) = \dfrac{T_0}{1 + \ln g}$ với g là bước chuyên hiện tại và T_0 là nhiệt độ ban đầu

7 Kết luận

Trong bài báo này chúng tôi đã sử dụng hai kỹ thuật hàng đầu về tìm kiếm toàn cục trong việc cải thiện giải thuật học của mạng nơron truyền thẳng nhiều lớp. Những nghiên cứu chính trong phạm vi của bài viết mới chỉ dừng ở mức độ lý thuyết. Chúng tôi hy vọng sẽ có điều kiện chứng minh tính hữu ích của phương pháp này trong những ứng dụng cụ thể. Xa hơn nữa chúng tôi cũng mong muốn được áp dụng các ý tưởng tích hợp trên một số các phương pháp khác trong lĩnh vực tính toán mềm như logic mờ, lập luận xấp xỉ ... nhằm nâng cao khả năng ứng dụng của chúng trong các bài toán thực tiễn về xử lý dữ liệu.

Tài liệu tham khảo

[1] **David E. Goldberg**: Genetic Algorithms in Search, Optimization, and Machine Learning, Reading, MA: Addison-Wesley(1989).

[2] **C.T. Lin, C.S.G. Lee**: Neurofuzzy, Academic Press, New York (1993).

[3] **LiMin Fu**: Neural networks in computer intelligence, McGraw-Hill, NewYork(1994).

[4] **L. R. Medsker**: Hybrid Intelligent Systems. Kluwer Academic Pub. (1995).

[5] **Trần Ngọc Hà, Nguyễn Thanh Thuỷ**: Mạng nơron nhiều lớp lan truyền ngược dùng cho việc mô hình hoá, Journal of computer science and cybernetics T.14.S.1.(1998),(26-33)45

[6] **J. Zupan, J. Gasteiger**: Neural networks for chemists, VCH Pub. (1993)

10 SUY RỘNG TOÁN TỬ OWA CỦA YAGER VÀ ỨNG DỤNG VÀO XỬ LÝ THÔNG TIN TRONG CÁC HỆ TRI THỨC

Bùi Công Cường
Viện Toán học

Nội dung

- Vào đầu.
- Phần 1: Toán tử trung bình trọng số có sắp xếp.
- Phần 2: Toán tử tích hợp ngôn ngữ.
- Phần 3: Một số ứng dụng.

Vào đầu

Năm 1988 R. Yager trong [1] đã định nghĩa toán tử trung bình trọng số có sắp xếp (ordered weighted averaging operators –OWA). Tiếp theo tác giả cùng nhiều nhà nghiên cứu đã trình bày hàng loạt khả năng sử dụng những toán tử này vào các bài toán khác nhau, ví dụ xem [2-4]. Gần đây, 1996 F.Herrera và các cộng sự [6-9] đã đưa vào lớp toán tử tích hợp ngôn ngữ dựa vào OWA và bắt đầu ứng dụng trong các bài toán quyết định tập thể. Theo hướng nghiên cứu này, từ 1998 chúng tôi cũng đã thu được một số kết quả [10-13]. Trong bài báo này sau phần giới thiệu những kết quả chính sẽ đi sâu thêm một số khía cạnh mới.

Phần 1: Toán tử trung bình trọng số có sắp xếp (OWA)

1 Định nghĩa và một số tính chất

Quá trình tích hợp thông tin xuất hiện trong rất nhiều ứng dụng của các hệ tri thức, ví dụ tích hợp trong mạng nơron (neural networks), điều khiển mờ, hệ chuyên gia và hệ trợ giúp quyết định, đặc biệt trong các bài toán phải xử lý những thông tin bất định. R. Yager [1] đã giới thiệu một kỹ thuật tích hợp mới đặt cơ sở trên toán tử trung bình có sắp xếp (ordered weighted averaging, viết tắt là OWA). Toán tử OWA này chỉ định nghĩa trên các vec tơ số thực, tuy nhiên như chúng ta sẽ thấy toán tử này có thể suy rộng để phát huy thế mạnh của nó trong các hệ tri thức.

Toán tử OWA, thực chất đã cung cấp các toán tử kết hợp (loại phép toán lấy trung bình) nằm giữa 2 phép toán logic "phép tuyển - OR" và "phép hội - AND".

1.1 Định nghĩa

Cho vectơ trọng số $w=[w_1, w_2, \ldots, w_n]^T$, các trọng số w_i thoả $0 \leq w_i \leq 1$, với mỗi $i=1, \ldots, n$ và thoả điều kiện $\sum_i w_i = 1$.

Định nghĩa 1.1: Cho vectơ $a=(a_1, \ldots, a_n) \in R^n$. Toán tử OWA là một ánh xạ $F: R^n \to R$, xác định bởi

$F(a) = \sum_j w_j b_j$.

trong đó b_j là phần tử lớn thứ j của vectơ a.

Ví dụ: Giả sử $W=[0.4, 0.3, 0.2, 0.1]^T$, $a=(0.7, 1, 0.3, 0.6)$, vec tơ b sẽ là $b=(1, 0.7, 0.6, 0.3)$ và $F(a) = (0.4)(1)+(0.3)(0.7)+(0.2)(0.6)+(0.1)(0.3) = 0.76$.

Ý cơ bản của toán tử này là sắp xếp lại, nghĩa là phần tử cần tích hợp a_i không kết hợp với trọng số w_i mà một trọng số sẽ kết hợp với một vị trí tương ứng của tập các phần tử tích hợp sau khi đã được sắp xếp.

Sự khác nhau giữa các toán tử OWA được phân biệt bởi các trọng số này.

Để minh hoạ, chúng ta xét một số trường hợp đặc biệt.

1) F^*: trong trường hợp $w = w^* = [1, 0, \ldots, 0]^T$.
2) F_*: ứng với $w = w_* = [0, \ldots, 0, 1]^T$.
3) F_{Ave}: tương ứng với $w = w_{Ave} = [1/n, \ldots, 1/n]^T$.

Dễ dàng thấy rằng:

$F^*(a_1, \ldots, a_n) = \max_i(a_i)$.

$F_*(a_1, \ldots, a_n) = \min_i(a_i)$.

và $F_{Ave}(a_1, \ldots, a_n) = \frac{1}{n}\sum_i a_i$.

1.2 Một số tính chất của toán tử OWA

1) Đối với mỗi toán tử OWA-F

$F_*(a_1, \ldots, a_n) \leq F(a_1, \ldots, a_n) \leq F^*(a_1, \ldots, a_n)$.

Do vậy điểm thấp nhất hay cao nhất của vectơ a là giới hạn của nó. Rõ ràng:

$\min_i(a_i) \le F(a_1, \dots, a_n) \le \max_i(a_i)$.

2) Tính hoán vị (*Commutative*)

Đặt $(a_1, \dots, a_n)$ là một gói cần tích hợp, và $(d_1, \dots, d_n)$ là một trong các hoán vị (*permutation*) của a_i. Khi ấy

$F(a_1, \dots, a_n) = F(d_1, \dots, d_n)$.

3) Tính đơn điệu (*monotonicity*)

Cho a_i và c_i là các phần tử được chọn lựa của tích hợp. Nếu $a_i \ge c_i$, với mỗi $i = 1, \dots, n$, thì:

$F(a_1, \dots, a_n) \ge F(c_1, \dots, c_n)$.

trong đó toán tử OWA–F có trọng số w cố định.

4) Tính luỹ đẳng (*idempotancy*)

Nếu $a_i = a$ với mọi i thì $F(a_1, \dots, a_n) = a$.

Từ 4 tính chất trên chúng ta thấy toán tử OWA có những tính chất kết hợp cơ bản như một toán tử trung bình.

1.3 Hai độ đo quan trọng gắn với toán tử OWA

Định nghĩa 1.2: *Độ phân tán hay entropy* (*dispersion or entropy*) của vector w được xác định bởi $\mathrm{Disp}(w) = -\sum_i w_i \ln w_i$.

Khi sử dụng toán tử OWA như là một toán tử trung bình thì $\mathrm{Disp}(w) = \ln(n)$ chính là mức độ khi ta sử dụng tất cả các phần tử tích hợp bằng nhau, khi đó giá trị của độ đo là lớn nhất. Còn trường hợp F^*, F_*, thì độ phân tán các trọng số là thấp nhất bằng 0.

Định nghĩa 1.3: *Độ đo tính tuyển* (*orness*) *và độ đo tính hội* (*andness*)

a) Độ đo tính tuyển của một vector w được cho bởi

$$\mathrm{orness}(w) = \frac{1}{n-1}\sum_{i=1}^{n}(n-i)w_i .$$

Có thể thấy $\mathrm{orness}(w^*) = 1$, $\mathrm{orness}(w_*) = 0$, $\mathrm{orness}(w_{Ave}) = 0.5$.

b) Độ đo tính hội của vector w được định nghĩa:

$\mathrm{andness}(w) = 1 - \mathrm{orness}(w)$.

Nói chung, các độ đo này sẽ đánh giá toán tử OWA với nhiều trọng **số gần đỉnh hơn sẽ** là toán tử "orlike" (giống toán tử tuyển "or"), và khi đó orness(W) ≥ 0.5.

Khi các trọng số là khác 0 và gần với đáy thì toán tử OWA được gọi là **giống toán tử hội** "andlike", khi đó orness(w) ≤ 0.5.

Định lý sau đây minh hoạ đặc tính này.

Định lý 1.4: Nếu w và w' là hai vector OWA có n thứ nguyên và với $\Delta > 0$, $i \neq j$, $j < k$ có:

a) $w_1 = w_1'$,

b) $w_j = w_j' + \Delta \quad (i \neq j, \Delta > 0)$,

c) $w_k = w_k' - \Delta \quad (\Delta > 0, j < k)$,

thì orness(w) > orness(w').

Chứng minh:

$$\text{orness}(w) = \frac{1}{n-1}\sum_{i=1}^{n}(n-i)w_i = \frac{1}{n-1}\left[\sum_{i=1}^{n}(n-i)w_i' + (n-j)\Delta - (n-k)\Delta\right]$$

$$= \text{orness}(w') + \frac{1}{n-1}\Delta(k-j).$$

Vì $k > j$ nên orness(w) > orness(w').

Định lý này chỉ ra rằng nếu chúng ta chuyển trọng số của vector w lên thì chúng ta sẽ làm tăng orness, khi chuyển trọng số xuống thì làm giảm orness(w).

2 Đối ngẫu của toán tử OWA

2.1 Định nghĩa 2.1: $\tilde{F}$ gọi là đối ngẫu (dual) của toán tử OWA–F, nếu nó là một toán tử OWA cùng thứ nguyên với trọng số $\hat{w}_i = w_{n+i+1}$.

Có thể dễ dàng thấy rằng nếu F và $\tilde{F}$ là một cặp đối ngẫu thì:

1) Disp($\tilde{F}$) = Disp(F)

2) orness($\tilde{F}$) = 1– orness(F)

Do vậy nếu F là giống tuyển (orlike) thì $\tilde{F}$ là giống hội (andlike.)

Bây giờ chúng ta hãy xét vài thay đổi khác của tập các trọng số OWA.

Giả sử F có trọng số w_i và $\hat{w}_i = (1 - w_i)/(n-1)$

Ví dụ: $w = [1\ 0\ 0\ 0\ 0]^T$ thì $\tilde{W} = [0\ 1/4\ 1/4\ 1/4\ 1/4\ ]^T$

Trong biến đổi này chúng ta làm cho mỗi w_i phân tán đều trong khoảng $n-1$ vị trí còn lại ở dạng các trọng số của vector mới.

Nếu $w_i = 1/n$ thì $\tilde{w}_i = (1-1/n)/(n-1) = 1/n$

Khi đó

$$\text{orness}(\tilde{F}) = \frac{1}{n-1}\sum_i (n-1)\tilde{w}_i = \frac{1}{n-1}\sum_{i=1}^{n}(n-i)\frac{1-w_i}{n-1}$$

$$= \frac{1}{(n-1)^2}\sum(n-i) - \frac{1}{(n-1)^2}\sum(n-i)w_i$$

$$= \frac{1}{(n-1)^2}\sum_i(n-i) - \frac{1}{n-1}\text{orness}(F)$$

$$= \frac{n(n-1)}{2(n-1)^2} - \frac{1}{n-1}\text{orness}(F)$$

$$= \frac{n}{2(n-1)} - \frac{1}{n-1}\text{orness}(F)$$

$$= \frac{1}{n-1}\left(\frac{n}{2} - \text{orness}(F)\right)$$

Kết quả của phép biến đổi này là làm xuất hiện việc đưa các trọng số đến gần với độ trung bình thực, $w_i = 1/n$. Phép toán này là một loại tuyến tính hoá của toán tử OWA.

2.2 Độ trội (Buoyancy measure)

Định nghĩa 2.2: Cho F là một toán tử OWA với trọng số w.

a) F có độ trội nếu các trọng số thoả mãn điều kiện: $w_i \ge w_j$ với mọi $i < j$.

b) F có độ trội mở rộng (*buoyancy measure extensive*) nếu như điều kiện là chặt theo nghĩa $w_i > w_j$ với mọi $i < j$.

Định lý 2.3: Nếu F là có độ trội thì $\text{orness}(F) \ge 0.5$.

Chứng minh: Ta có

$$\text{orness}(F) = \frac{1}{n-1}\sum_{i=1}^{n}(n-1)w_i = \frac{1}{2} + \frac{1}{n-1}\sum_{i=1}^{n}(n-1)w_i - \frac{1}{2}$$

mà $\sum_i w_i = 1$, vậy $\frac{1}{2}\sum_i w_i = \frac{1}{2}$, nên:

$$\text{orness}(F) = \frac{1}{2} + \frac{1}{n-1}\sum_{i=1}^{n}(n-1)w_i - \frac{1}{2}\sum_i w_i .$$

$$= \frac{1}{2} + \sum_i\left(\frac{n-i}{n-1} - \frac{1}{2}\right)w_i = \frac{1}{2} + \sum_i\left(\frac{n-2i+1}{2(n-1)}\right)w_i .$$

Đặt $q_i = \dfrac{n-2i+1}{2(n-1)}$, $i = 1, \ldots, n$.

nên $q_{i+1-i} = \dfrac{n-2n-2+2i+1}{2(n-1)} = \dfrac{-n+2i-1}{2(n-1)} = -q_i$.

Trường hợp n chẵn, ta có $n = 2m$ và khi ấy $i = a$, $a \le m$

$$\text{orness}(F) = \frac{1}{2} + \sum_{a=1}^{m} q_a (w_a - w_{n+1-a}) \ge \frac{1}{2}$$

Vì: $q_a = \dfrac{n-2a+1}{2(n-1)} \ge \dfrac{2m-2m+1}{2(n-1)} \ge 0$ và $w_i \ge w_j$ với $i < j$.

Trường hợp n là số lẻ, tức là $n = 2m+1$ chúng ta có:

$$\text{orness}(F) = \frac{1}{2} + \sum_{a=1}^{m} q_a (w_a - w_{n+1-a}) + q_{m+1} w_{m+1}$$

mà: $q_{m+1} = \dfrac{2m+1-2(m+1)+1}{2(n-1)} = 0$

nên $\text{orness}(F) \ge \dfrac{1}{2}$.

Bổ đề 2.4: Nếu trọng số thoả điều kiện $w_i \ge w_{n-i+1}$ thì orness(F) ≥ 0.5.

Nếu $w_i \le w_{n-i+1}$ thì orness(F) ≤ 0.5.

Một lớp quan trọng của độ trội là độ trội mạnh (*strong buoyancy measure*).

Một toán tử OWA n thứ nguyên được gọi là có độ trội mạnh nếu:

$$w_i \ge \sum_{j=i+1}^{n} w_i \quad \text{với } i = 1, \ldots, n-1.$$

3 Ngữ nghĩa kết hợp với toán tử OWA

Một số ngữ nghĩa có thể được kết hợp với việc sử dụng toán tử OWA (xem [2]). Đó là các ngữ nghĩa gắn bó với một số ứng dụng của toán tử này.

3.1 Ngữ nghĩa đầu tiên và rất tự nhiên được kết hợp với toán tử tích hợp OWA là một loại toán tử trung bình.

Ở đây độ phân tán Disp(w) chỉ ra lượng biến thông tin được sử dụng trong quá trình tính trung bình.

Với phép lấy trung bình quen biết thì orness(w) = 0.5, do đó chúng ta có thể sử dụng độ đo orness để xác định độ chệch khỏi trung bình:

$$\text{Bias}(w)= \frac{1}{2}(\text{orness}(w) - \frac{1}{2})$$

Từ định nghĩa của Bias(w), cho ta nhận xét sau:

Bias(w) > 0: giá trị cao hơn được nhấn mạnh

Bias(w) < 0: giá trị thấp hơn được nhấn mạnh

Bias(w) = 0: không nhấn mạnh giá trị nào

Độ chệch này cùng với giá trị độ phân tán cho chúng ta một hình ảnh tốt về toán tử được thể hiện.

3.2 R. Yager cho rằng có thể sử dụng các trọng số của toán tử OWA như là một loại phân phối xác suất đặc biệt.

Do $w_i \in [0,1]$ và $\sum_i w_i = 1$, rõ ràng trọng số của toán tử OWA có những tính chất của một phân phối xác suất.

Giả sử chúng ta có một quyết định cần đưa ra. Ứng với mỗi phương án lựa chọn có một tập các kết quả xảy ra $A = \{a_1, a_2, \ldots, a_n\}$. Mà sự thực, một vài phần tử trong A được xác định bởi một số tác nhân ngoại sinh được gọi là tự nhiên.

Trên A xác định phân phối xác suất P, với p_i là xác suất mà kết quả thứ i xảy ra tốt nhất.

Gọi V là giá trị của một phương án thì:

$$V = F(a_1, a_2, \ldots, a_n)$$

Trong đó F là một toán tử OWA với trọng số $w_i = p_i$

Nếu $w_n = p_n = 1$ thì giả sử rằng khả năng tồi nhất sẽ xảy ra trong "tự nhiên hiểm ác". Sự lựa chọn này của toán tử OWA chính là kĩ thuật quyết định Max, Min.

Trong môi trường lấy quyết định, chúng ta sẽ thấy rằng độ phân tán có thể được hiểu như entropi của phân phối xác suất. Hơn nữa, độ orness của W có thể như là độ lạc quan của quyết định, trong đó độ andness là chỉ tiêu về độ bi quan.

Nếu $w_i = p_i = \alpha$ và $w_n = p_n = 1 - \alpha$, ta có:

$$\text{optimism}(w) = \frac{1}{n-1}\sum_{i=1}^{n}((n-1)w_i = \frac{1}{n-1}((n-1)\alpha) + \frac{n-n}{n-1}(1-\alpha) = \alpha$$

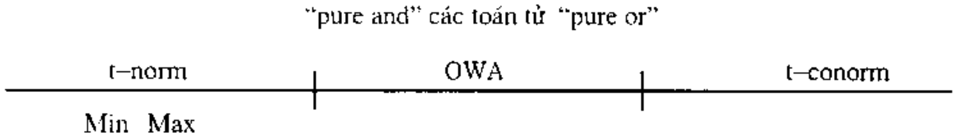
3.3 Một vùng khác toán tử OWA được sử dụng rất thành công là trong lý thuyết tập mờ và logic đa trị. Chúng ta thấy rằng rất nhiều các phép toán của tập mờ sử dụng trong các ứng dụng có các phép toán sử dụng logic đa trị (*multivalued logic*).

Nếu A và B là 2 tập mờ của x, ta có với mỗi $x \in X$, thì độ thuộc (*membership degrees*) $A(x)$, $B(x)$ có thể được xác định bởi những số trong khoảng $[0,1]$. Khi đó phép giao $E=A \cap B$ và phép hợp $F=A \cup B$ được định nghĩa bằng:

$$E(x)=T(A(x),B(x)), \qquad F(x)=S(A(x),B(x)),$$

trong đó T và S là cặp toán tử t–chuẩn và t–đối chuẩn. Các toán tử này là họ toán tử cần thiết của logic đa trị "and" và "or". Về các toán tử này xem thêm trong [14, 15].

Vậy Min là "pure and"(phép hội thuần tuý) lớn nhất và Max là "pure or" (phép tuyển thuần tuý) nhỏ nhất. Hình dưới đây sẽ chỉ ra mối liên hệ giữa toán tử OWA và t–chuẩn (t–corm) và t–đối chuẩn (t–cornom).



Hình 1: Mối liên hệ giữa toán tử OWA, t-chuẩn và t-đối chuẩn

Chúng ta thấy rằng toán tử OWA cung cấp sự biến đổi liên tục từ "pure and" tới "pure or". Trong quá trình này orness đo mức độ toán tử là 'orlike' hoặc 'andlike'. Nếu giá trị orness lớn hơn 0.5, toán tử đó là 'orlike', còn nếu nhỏ hơn 0.5 thì toán tử đó là 'andlike'.

Ví dụ: $w=[0\ \ 0\ \ 0.1\ \ 0.2\ \ 0.7]^T$ sẽ là phép toán 'andlike'.

Đối với toán tử OWA có $w_i=1/n$, thì orness(w)=0.5. Khi đó chúng ta thấy trung bình thực sự như ranh giới giữa phép toán 'andlike' và 'orlike'.

3.4 Một đặc trưng nữa của toán tử OWA là đóng một vai trò quan trọng trong những mô hình lượng hoá ngôn ngữ.

Theo logic cổ điển có 2 lượng hoá quen thuộc, đó là " có tồn tại một - there exits" và "với mọi - for all ".

Trong ngôn ngữ tự nhiên chúng ta sử dụng rất nhiều lượng hoá khác "mờ hơn" như là "hầu hết – almost all", "một ít – few", "đa số – most", "gần một nửa – nearly half".

Zadeh trong [19] đã cho rằng việc lượng hoá tất cả những gì chúng ta vẫn dùng trong ngôn ngữ thông thường đều có thể làm được bằng những tập mờ trong một khoảng nào đó.

Do vậy, nếu Q là một lượng tử, ví dụ " đa số – most " thì Q có thể được thể hiện dưới một tập mờ của L trong đó với mỗi $r \in L$, thì $Q(r)$ sẽ chỉ ra mức độ của vị trí r được thoả

mãn bởi nội dung định nghĩa ở Q, do vậy nếu Q là most thì $Q(0.8)=1$ chúng ta sẽ nói rằng 80% là hoàn toàn thoả mãn với ý định đưa ra bởi sự lượng hoá 'most'.

4 Cách xác định trọng số w

Chúng ta đều nhận thấy việc chọn vectơ w có ý nghĩa rất quan trọng. Sau đây sẽ giới thiệu nhanh 2 cách xác định vectơ trọng số.

4.1 Xác định qua các lượng tử mờ Q.

Hãy xét bài toán thực tiễn sau: Cho tập phương án X và một tập hữu hạn các tiêu chuẩn (ví dụ: đất, rẻ, bền, khả thi, độ may rủi ...). Những tiêu chuẩn này được biểu diễn như là các tập mờ trên X, với độ thuộc $A_i(x)$, theo tiêu chuẩn thứ i.

Bây giờ hãy đánh giá giá trị chân lý $v(P)$ của mệnh đề mờ P sau:

P = "x thoả mãn đa số các tiêu chuẩn".

R. Yager từ sớm đã đề nghị thuật toán tính $v(P)$ với mỗi phương án x.

Thuật toán 4.1: Thuật toán đánh giá $v(P)$.

Cố định phương án x. Thực hiện các bước:

1) Xác định hàm định lượng " đa số - most " bằng một hàm Q đơn điệu không giảm trên $[0,1]$, thoả điều kiện $Q(0)=0$, $Q(1)=1$.
2) Với mỗi tiêu chuẩn A_i tính $a_i = A_i(x)$, $i=1, ..., n$.
3) Xác định $w_i=Q(i/n)-Q((i-1)/n)$.
4) Đánh giá $v(P)$ của mệnh đề P bằng sử dụng toán tử OWA với vectơ a và w vừa xác định.

4.2 Học trọng số w từ dữ liệu.

Filev và Yager trong [3] đã giới thiệu một thuật toán học trọng số w từ mẫu dữ liệu. Kỹ thuật này chịu ảnh hưởng của các thuật học trong mạng nơron nhân tạo và cũng dựa vào phương pháp tụt gradient.

Giả sử chúng ta có m mẫu dữ liệu dạng vec tơ n chiều

$a_k=(a_{k1}, a_{k2}, \dots, a_{kn})$ và một giá trị gộp d_k, với $k=1, \dots, m$.

Mục dích của chúng ta là sử dụng toán tử OWA để mô hình hoá vấn đề gộp này. Vấn đề trở thành bài toán xác định vectơ w thích hợp – vậy đây là bài toán học trọng số w từ mẫu dữ liệu.

Gọi $b_k=(b_{k1},b_{k2}, \ldots ,b_{kn})$ là vectơ dữ liệu đã sắp xếp. Hãy tìm $w=(w_1,w_2, \ldots ,w_n)$ sao cho

$$b_{k1}w_1+ b_{k2}w_2+ \cdots +b_{kn}w_n=d_k \quad \text{với} \quad k=1, \ldots ,m.$$

Phương pháp giải là phép lặp sao cho đạt min sai số

$$e_k = \frac{1}{2}(b_{k1}w_1+ b_{k2}w_2+ \cdots +b_{kn}w_n-d_k)^2.$$

Thuật toán 4.2: Thuật toán được tính lặp theo t.

- **Bước 1:** Tại bước lặp t chúng ta đang ở giá trị tham số u, $u_i(t)$ và một quan sát mới $b_k=(b_{k1},b_{k2}, \ldots ,b_{kn})$ cùng với giá trị gộp d_k.
- **Bước 2:** Ước lượng:

  $w_i(t) = \exp(u_i(t)) / \sum_k \exp(u_k(t))$
- **Bước 3:** Tính giá trị gộp d_k^*:

  $d_k^*= b_{k1}w_1(t)+ b_{k2}w_2(t)+ \cdots +b_{kn}w_n(t)$
- **Bước 4:** Điều chỉnh tham số u_i:

  $u_i(t+1)=u_i(t)-\beta w_i(t)(b_{ki} - d_k^*)(d_k^* - d_k)$,

  ở đây β là hằng số học.

5 Các hàm định lượng và độ đo tính tuyển – orness

5.1 Bây giờ chúng ta sẽ làm việc với lớp các định lượng sau:

Định nghĩa 5.1: Q là một hàm định lượng thông thường, đơn điệu, không giảm nếu thoả các điều kiện:

$Q(0)=0$; $Q(1)=1$ và $Q(r_1)\geq Q(r_2)$ nếu $r_1>r_2$.

Sau đây là một vài kết quả:

Định lý 5.2: Nếu toán tử OWA F_w được tạo nên từ một hàm định lượng không giảm bình thường Q với $w_i=Q(i/n)-Q((i-1)/n)$, thì

orness(w) = $\{\sum_i Q(i/n): i=1, \ldots ,n-1\} / (n-1)$.

Chứng minh: Thật vậy

$$\text{orness}(w) = \frac{1}{n-1}\sum_{i=1}^{n}(n-i)w_i$$

$$= (1/(n-1))\{-(n-1)Q(0/n)+Q(1/n)+Q(2/n)+ \cdots +Q((n-1)/n)\}.$$

Chúng ta có kết quả vì $Q(0)=0$.

Định lý 5.3: Giả sử Q_1, Q_2 là hai hàm lượng hoá sao cho $Q_1(x) \le Q_2(x)$. Khi ấy

$\text{orness}(F_{Q_1}) \le \text{orness}(F_{Q_2})$.

Hệ quả 5.4: Giả sử Q là một hàm lượng hoá, khi đó

a) Nếu $Q(r) \ge t$ với mọi r, thì Q là "orlike" và $\text{orness}(F_Q) \ge 0.5$

b) Nếu $Q(r) \le t$ với mọi r, thì Q là "andlike" và $\text{orness}(F_Q) \le 0.5$

Tương tự, chúng ta có mệnh đề sau:

Định lý 5.5: a) Họ các hàm lượng hoá $Q(r^t)$, $t > 1$ sinh ra họ các lượng hoá "andlike". t càng lớn tính hội (andlike) càng lớn.

b) Họ các hàm lượng hoá $Q(r^t)$, $t < 1$ sinh ra họ các lượng hoá "orlike". t càng bé tính tuyển (orlike) càng bé.

Phần 2: Toán tử tích hợp ngôn ngữ

1 Cần một suy rộng lên miền giá trị ngôn ngữ

1.1 Chúng ta sẽ xét các lớp toán tử tích hợp cho giá trị trên tập từ như vẫn thường dùng trong ngôn ngữ đời thường. Để dễ hình dung chúng ta xét bài toán đánh giá các dự án sau.

1.2 Các chỉ tiêu định tính và việc đánh giá cho bằng từ.

Thông thường khi xem xét, đánh giá các dự án trước tiên người ta quan tâm tới một số chỉ tiêu định lượng. Ví dụ chỉ tiêu: Tổng vốn đầu tư, Thời gian hoàn vốn. Hay như các chỉ tiêu thường được nhắc trong các bài giảng về quản lý dự án như: Tỷ suất nội hoàn IRR (Internal Rate of Return).

Bên cạnh các chỉ tiêu định lượng, chẳng hạn với các dự án công nghệ thông tin, người ta vẫn thường xuyên nhắc tới một số chỉ tiêu định tính như: Độ may rủi (Potential Risk), Tính khả thi (Feasibility), Độ tương thích (Suitability), v.v.

Đã có những Hội đồng mong muốn các cố vấn cho đánh gía bằng số về các chỉ tiêu định tính này. Chẳng hạn họ muốn các chuyên gia phát biểu dưới dạng: 'độ khả thi của dự án A4 là 35%', hay 'độ may rủi của dự án A2 là 25%'. Đó là một mong muốn chẳng thể nào thực hiện được một cách nghiêm túc.

Một cách tiếp cận khoa học, khách quan, tương đối dễ thực hiện là để các cố vấn – chuyên gia phát biểu bằng từ như vẫn dùng trong ngôn ngữ thông thường.

Ví dụ với chỉ tiêu 'Độ may rủi ' có thể chọn tập nhãn S sau đây để các chuyên gia lựa chọn phát biểu:

S = {*hầu như không* , *rất thấp* , *thấp* , *trung bình* , *cao* , *khá cao* , *rất cao* }.

Cùng với các cách tính toán truyền thống thông qua các công thức, các mô hình chặt chẽ, rõ ràng, cần bổ sung các cách thu thập và tính toán với thông tin từ nhiều nguồn, đặc biệt từ thu thập ý kiến đánh giá, kinh nghiệm của các cố vấn – chuyên gia. Chấp nhận và tổ chức tốt để thu nhận được những đánh giá cho bằng từ vẫn dùng trong ngôn ngữ thông thường.

1.3 Cơ sở dữ liệu và cơ sở tri thức

Đây là hai phân hệ không thể thiếu trong các hệ hỗ trợ ngày nay.

Chúng tôi tạm dừng lại mấy ý trong việc xây dựng cơ sở tri thức cho hệ hỗ trợ đánh giá các dự án.

- Cố gắng làm việc nghiêm túc với các chuyên gia giỏi, am hiểu và có kinh nghiệm thực tiễn về lĩnh vực đang quan tâm.
- Hệ cần được thiết kế bởi những người có tri thức vững vàng về khoa học hệ thống, phân tích hệ thống (đáng tiếc các kỹ sư chúng ta đào tạo ra ngày nay rất yếu phần này).
- Trong cơ sở tri thức nên tổ chức thu thập tốt các tri thức (hiểu theo nghĩa rộng) quan trọng nhất. Dành sự quan tâm đúng mức tới các luật (rules) dạng quan hệ nhân quả. (Vì ngày nay khả năng xử lý các bộ luật đó đã tiến bộ rất nhiều – những ví dụ đơn giản có thể xem trong [17, 18])
- Lựa chọn cẩn thận các chỉ tiêu định lượng và các chỉ tiêu định tính. Với các chỉ tiêu định lượng nên xem xét kỹ mô hình và cách tính tóan. Với các chỉ tiêu định tính cố gắng phân tích sâu hơn, chẳng hạn thông qua các chỉ tiêu định tính khác từng bước chi tiết hơn.

Ví dụ để ước lượng 'độ may rủi' của các dự án công nghệ thông tin ta xét tới bộ 3 các chỉ tiêu sau:

- Độ phức tạp của dự án.
- Thời gian phát triển dự án.
- Các phản ứng cạnh tranh trên thương trường.

1.4 Thu thập đánh giá, xử lý thông tin và phân lớp các dự án

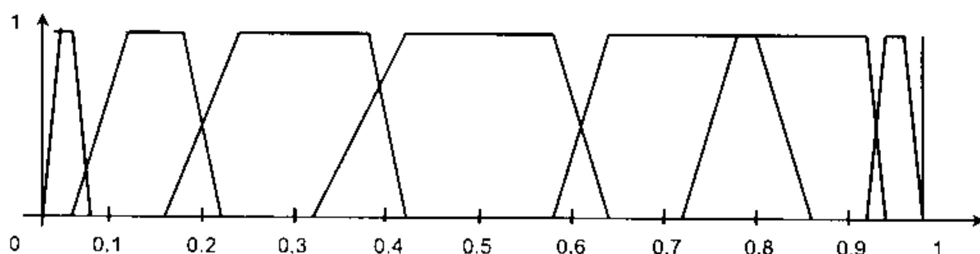
Sau đây chúng tôi tập trung trình bày một mô hình chính, phục vụ hỗ trợ cho nhiệm vụ đã chọn.

Mô hình chính:

Cho $A = \{A_1, A_2, \dots, A_n\}$ là n dự án cần đánh giá và phân lớp.

Cho $E = \{e_1, e_2, \dots, e_m\}$ là m chuyên gia – cố vấn tham gia Hội đồng đánh gía.

Tạm thời chúng ta cố định một chỉ tiêu. Giả sử tập nhãn bằng từ cho dưới dạng sau:



Hình 2: Đồ thị hàm liên thuộc nhãn đánh giá.

Ví dụ: Tập nhãn S sau đây gồm 9 từ có thể sử dụng cho các chuyên gia đánh giá về các kết luận dạng:

Q = "Dự án A_3 này rất khả thi"

hay các kết luận có dạng

Q = "Dự án A_1 tốt hơn, hiệu quả hơn dự án A_3".

S = "$s_1, s_2, \dots, s_9$"

chúng ta có thể chọn từ tiếng Việt tương ứng là

S = "*không thể xẩy ra , không có khả năng , rất ít khả năng , ít khả năng , có thể, có nhiều khả năng , rất có khả năng , hoàn toàn có khả năng , hiển nhiên*".

(Biểu diễn những từ này dưới dạng các số mờ có thể tìm thấy trong [6]).

C	Certain	Hiển nhiên	(1, 1, 1, 1)
EL	Extremely_likely	Hoàn toàn có khả năng	(0.93, 0.98, 0.99, 1)
ML	Most_likely	Rất có khả năng	(0.72, 0.78, 0.92, 0.97)
MC	Meaningful_chance	Có khả năng	(0.58, 0.63, 0.80, 0.86)
IM	It_may	Có thể	(0.32, 0.41, 0.58, 0.65)
SC	Small_chance	Có ít khả năng	(0.17, 0.22, 0.36, 0.42)

VLC	Very_slow_chance	Rất ít khả năng	(0.04, 0.10, 0.18, 0.23)
EU	Extremely_unlikely	Không có khả năng	(0.00, 0.01, 0.02, 0.07)
I	Impossible	Không thể xảy ra	(0. 0. 0, 0)

Hình 2 là đồ thị thể hiện các hàm liên thuộc các nhãn đánh giá:

2 Một suy rộng: toán tử tích hợp ngôn ngữ LOWA

2.1 Định nghĩa

Sử dụng khái niệm tổ hợp lồi của J.Delgado [5], F.Herrera và cộng sự trong [6, 7, 8] đã trình bày định nghĩa một lớp toán tử LOWA trực tiếp suy rộng toán tử OWA của R.Yager và áp dụng trong các bài toán quyết định tập thể. Tuy nhiên trong quá trình tìm cách ứng dụng định nghĩa trong [6, 7] vào trong bài toán đánh giá và ước lượng cá dự án công thức đã cho trong [6, 7] tỏ ra không phù hợp.

Với gợi ý đó, từ năm 1998 (xem [10, 12] tác giả đã sử dụng công thức tính dưới đây.

Cho $S=\{s_1,s_2, \dots ,s_T\}$ là tập nhãn (tập từ như đã giải thích), sắp toàn phần $s_1<s_2< \dots < s_T$. Cho $a=\{a_1,a_2, \dots ,a_m\}$ là tập các từ cần tích hợp, mỗi a_i nhận giá trị trong S. b là tập a đã sắp xếp $b=\{b_1,b_2, \dots ,b_m\}$, trong đó b_j là phần tử lớn thứ j của a. Như vậy $b=\{s_{im},s_{i(m-1)}, \dots , s_{i1}\}$ với $i_m \geq i_{m-1} \geq \dots \geq i_1$.

Cho $w=\{w_1,w_2, \dots ,w_m\}$ là vector trọng số, $w_i \in [0, 1]$ và $\sum_i w_i=1$.

Định nghĩa 2.1: Tổ hợp trọng số của a và w – $C\{a,w\}$ là tổ hợp trọng số thực (actual combination), nếu

$C\{a,w\} = C\{a',w'\}$,

ở đây a',w' là cái thu hẹp của a và w trên tập những chỉ số i nào mà $w_i > 0$.

Hệ quả 2.2: Nếu có một chỉ số i_0 sao cho $w_{i0} = 1$, đồng thời $w_i = 0$ với mọi $i \neq i_0$, thì mỗi tổ hợp thực $C\{a,w\} = a_{i0}$.

Nhận xét 2.3: Nếu chỉ quan tâm tới tổ hợp trọng số thực thì chúng ta có thể định nghĩa LOWA chỉ với các vectơ trọng số w có mọi $w_i > 0$.

Định nghĩa 2.4: Cho tập $a=\{a_1,a_2, \dots ,a_m\}$, $w=[w_1,w_2, \dots ,w_m]$ là vector trọng số. $w_i> 0$, với mỗi i. Toán tử LOWA là một tổ hợp thực của vectơ a với trọng số w. Low: $(a,w) \rightarrow S$ cho bởi công thức truy toán sau:

$\text{Low}(a, w) = C\{(w_{im}, a_{im}), (1- w_{im}, \text{Low}(a', w')\}$,

ở đây $a' = \{a_{i(m-1)}, \ldots, a_{i1}\}$, $w' = [w'_{i1}, w'_{i2}, \ldots, w'_{i(m-1)}]$, $w'_j = w_j / (1-w_{im})$, C là

phép tổ hợp của 2 nhãn (s_j, s_i), $j \geq i$ với trọng số $w_j > 0$, $w_i > 0$, $w_j + w_i = 1$,

$C\{(w_j, s_j), (w_i, s_i)\} = s_k$, với $k = i + \text{round}(w_j \cdot (j-i))$ (round là phép làm tròn số).

Nhận xét 2.5: Rõ ràng nếu tập S nhận các giá trị trên R^1 thì toán tử Low là phép lấy trung bình có trọng số quen biết, (do vậy $\text{Low}(a, w)$ sẽ là kỳ vọng toán học khi w là vectơ xác suất).

Ví dụ 2.6: Cho $a = (s_1, s_2, s_3)$. Cho $w = (0.2, 0.3, 0.5)$. Khi đó $b = (s_3, s_2, s_1)$, $w_3 = 0.5$, $w_2 = 0.3$, $w_1 = 0.2$ và

$$\text{Low}(a, w) = C\{(0.5, s_3), (0.5, \text{Low}((s_2, s_1), (0.2 / 0.5, 0.3/ 0.5).$$

Song

$$\text{Low}((s_2, s_1), (0.2 / 0.5, 0.3/ 0.5) = C\{(3/5, s_3), (2/5), s_2)\} = s_{k1}$$

$$k_1 = 1+\text{round}((3/5)\cdot(2-1)) = 1+1 = 2.$$

Do vậy

$$\text{Low}(a, w) = C\{(0.5, s_3), (0.5, s_2)\} = s_k, k = 2 + \text{round}(0.5 \cdot (3-2)) = 3.$$

Cuối cùng

$$\text{Low}(a, w) = s_3.$$

Ví dụ 2.7: Cho $S = (s_1, s_2, \ldots, s_9)$, $w = (0, 0, 0.2, 0, 0.3, 0.2, 0, 0.1, 0.2)$.

Khi đó $\text{Low}(S,w) = \text{Low}(a,w')$, ở đây $a = (s_1, s_2, \ldots, s_9)$, $w' = (0.2, 0.3, 0.2, 0.1, 0.2)$, $b = (s_9, s_3, s_6, s_5, s_3)$. Như vậy

$$\text{Low}(a,w') = C\{(0.2, s_9), (0.8, \text{Low}((s_8, s_6, s_5, s_3), (2/8, 3/8, 2/8, 1/8) = s_{k1}$$

và $$\text{Low}((s_8, s_6, s_5, s_3),(2/8,3/8,2/8,1/8)) = C\{(1/8,s_8),(7/8,\text{Low}(s_6,s_5,s_3),(2/7,3/7,2/7)\} = s_{k2}.$$

Tương tự, dùng công thức truy toán chúng ta có

$$\text{Low}(s_6, s_5, s_3),(2/7,3/7,2/7) = C\{(2/7,s_6),(5/7,\text{Low}((s_5, s_3),(2/5,3/5)\}$$
$$= C\{(2/7,s_6),(5/7,s_4)\} = s_5.$$

Do đó

$$s_{k2} = s_5, \quad \text{và} \quad s_{k1} = s_6.$$

Hay là

$$\text{Low}(a, w') = s_6.$$

Phần 3: Một số ứng dụng

1 Hai thuật toán phân cụm (clustering analysis algorithms)

Trước tiên chúng ta xét bài toán đánh giá các dự án, các phương án sau.

Mô hình:

– Cho $A=\{A_1, A_2, \ldots, A_n\}$ là n dự án cần đánh giá và phân lớp.

– Cho $E=\{e_1, e_2, \ldots, e_m\}$ là m chuyên gia – cố vấn tham gia Hội đồng đáng gía.

– Cho $W=\{w(1), w(2),\ldots, w(k),\ldots,w(m)\}$. $w(k)$ là trọng số của chuyên gia e_k, $0\leq w(k)\leq 1$.

Chọn tương ứng một tập nhãn S bằng từ để các chuyên gia lựa chọn và thực hiên đánh giá các dự án. Ví dụ

$S=$ "$s_1, s_2, \ldots, s_9$"

chúng ta có thể chọn từ tiếng Việt tương ứng là

$S=$ "*không thể xẩy ra , không có khả năng , rất ít khả năng , ít khả năng , có thể, có nhiều khả năng , rất có khả năng , hoàn toàn có khả năng , hiển nhiên*".

1) *Thuật toán phân cụm* 1: Thuật toán dựa vào đánh giá trực tiếp của các chuyên gia.

Mỗi chuyên gia e_k cho đánh giá các dự án A_i bằng một phép đánh giá

$P_k: A \to S.$

Mọi thông tin đánh giá là $\{P_k: k=1, \ldots, m\}$.

Thuật toán 1

a) *Bước* 1. Thu thập các vectơ $\{P_k\}$.

b) *Bước* 2. Dựa vào vec tơ trọng số $W=\{w(1), w(2),\ldots, w(k),\ldots,w(m)\}$, $w(k)$ là trọng số của chuyên gia e_k, $0\leq w(k)\leq 1$, chuẩn hoá $w'(k)=w(k)/w_0$, ở đây $w_0=\Sigma_k w(k)$.

c) *Bước* 3. Tính trọng số gộp theo từng nhãn s_t đối với mỗi phương án A_i – đó chính là độ nhất trí chọn s_t của cả Hội đồng khi đánh giá A_i

$$\mathrm{IC}(i)[s_t] = \Sigma_k \{w'(k): P_k(A_i)=s_t\}$$

d) *Bước* 4. Tính độ trội của mỗi phương án A_i (cho bằng từ) bằng toán tử LOWA

$$E(i)=\mathrm{Low}(S, U(i)),$$

ở đây $U(i)=[u_1, \ldots, u_t, \ldots, u_T]$, với $u_t=\mathrm{IC}(i)[s_t]$, với mỗi t.

e) *Bước* 5. Phân cụm. Dùng $\{E(i): A_i \in A\}$ để phân cụm tập phương án A thành các lớp $Y_1, Y_2, \ldots, Y_T$, $Y_t = \{A_i E(i)=s_t\}$, với mỗi $s_t \in S$.

Rõ ràng $A=\cup_t Y_t$. Mỗi tập con Y_t có thể là tập rỗng, đối với những tập Y_t khác rỗng ta có thể sắp xếp theo sự sắp xếp của tập nhãn S như sau: $Y_t < Y_{t'}$ nếu $t < t'$.

2) *Thuật toán phân cụm* 2: Thuật toán dùng thông tin dạng đánh giá so sánh từng cặp của các chuyên gia.

Mỗi chuyên gia e_k cho đánh giá dạng so sánh dự án A_i với dự án A_j bằng một phép đánh giá $P_k : A\times A \rightarrow S$.

Mọi thông tin để phân cụm là thông tin đánh giá là $\{P_k: k=1, \ldots, m\}$. Để gọn ta kí hiệu $P_k(i,j)=P_k(A_i,A_j)$. Chúng ta sẽ giả thiết rằng $P_k(i,i)=s_{(T+1)/2}$, với mỗi i và nếu $P_k(i,j) \geq s_{(T+1)/2}$ thì $P_k(j,i) \leq s_{(T+1)/2}$.

Thuật toán 2

a) *Bước* 1. Thu thập các quan hệ mờ $\{P_k\}$.

b) *Bước* 2. Dựa vào vec tơ trọng số $W=\{w(1), w(2), \ldots, w(k), \ldots, w(m)\}$, $w(k)$ là trọng số của chuyên gia e_k, $0\leq w(k)\leq 1$, chuẩn hoá $w'(k)=w(k)/w_0$, ở đây $w_0=\Sigma_k w(k)$.

c) *Bước* 3. Tính trọng số gộp theo từng nhãn s_t đối với mỗi cặp phương án (A_i,A_j) – đó chính là độ nhất trí chọn s_t trong so sánh của cả Hội đồng.

$$\mathrm{IC}(i,j)[s_t] = \Sigma_k\{w'(k): P_k(A_i,A_j)=s_t\}$$

d) *Bước* 4. Tính độ trội tương đối của mỗi cặp phuơng án (A_i,A_j) (cho bằng từ) bằng toán tử LOWA

$$E(i,j)=\mathrm{Low}(S, U(i,j)),$$

ở đây $U(i,j)=[u_1, \ldots, u_t, \ldots, u_T]$, với $u_t=\mathrm{IC}(i,j)[s_t]$, với mỗi t.

Để ý mỗi $E(i,j)$ là một ma trận, như vậy đó là một quan hệ mờ cấp 2- quan hệ mờ này đo độ trội tương đối theo ý kiến đã tích hợp của cả Hội đồng.

e) *Bước* 5. Tìm nghiệm tập thể mờ FCS (the fuzzy collective solution xem [13]. Đây là một tập mờ loại 2 xác định trên cho bởi

$$\mathrm{FCS} = (\mu_{\mathrm{FCS}}(A_1)/A_1, \ldots, \mu_{\mathrm{FCS}}(A_n)/A_n),$$

với $\mu_{\mathrm{FCS}}(A_i) = \mathrm{Low}(S, V(i))$, $V(i)=[v_1, \ldots, v_t, \ldots, v_T]$,

với mỗi t, $v_t=\#\{j : E(i,j)=s_t, j\neq i\}/m-1$.

f) *Bước* 6. Phân cụm. Dùng $\{\mu_{FCS}(A_i): A_i \in A\}$ để phân cụm tập phương án A thành các lớp $Y_1, Y_2, \ldots, Y_T$ sao cho $Y_t = \{A_i\ \mu_{FCS}(A_i) = s_t\}$, với mỗi $s_t \in S$.

Rõ ràng $A = \cup_t Y_t$. Mỗi tập con Y_t có thể là tập rỗng, đối với những tập Y_t khác rỗng ta có thể sắp xếp theo sự sắp xếp của tập nhãn S như sau: $Y_t < Y_{t'}$, nếu $t < t'$.

2 Độ nhất trí và độ trội địa phương

Để thuận lợi cho trình bày hai quy trình chọn giải pháp được kiến nghị làm quyết định, chúng ta cần thêm hai độ đo sau.

1) *Độ đo sự nhất trí.*

Định nghĩa 3.1: a) Hàm lượng hoá đơn điệu Q_1 ứng với $[a, b] \subset [0, 1]$, $a < b$ cho bởi:

$$Q_1(z) = \begin{cases} 0 \text{ nếu } z < a \\ (z-a)/(b-z) \text{ nếu } a \le z \le b \\ 1 \text{ nếu } z > b \end{cases}$$

b) Hàm lượng hoá ngôn ngữ Q_2 tương ứng với Q_1 cho bởi:

$$Q_2(z) = \begin{cases} s_1 \text{ nếu } z < a \\ s_i \text{ nếu } a \le z \le b \\ s_T \text{ nếu } z > b \end{cases}$$

ở đây $s_i = \max\{s_1 : s_1 \in M\}$,

$M = \{ s_1 : \mu_1(z) = \max \{\mu_t(z - a/b - a) : s_1 \in S\}$.

Độ đo nhất trí sau đây áp dụng với mô hình đánh giá trong thuật toán 2 vừa trình bày.

Giả sử Q_2 là hàm lượng hóa ngôn ngữ phản ánh quan niệm về đám đông (tập thể).

Định nghĩa 3.2: Cho hàm Q_2.

a) Với mỗi cặp (A_i, A_j) độ đo nhất trí về cặp (i, j) cho bởi

$\text{PM}(i, j) = Q_2(\text{IC}(i, j))$, với $\text{IC}(i, j) = \max\{\text{IC}(i, j)[s_t] : s_t \in S\}$.

b) Với mỗi phương án A_i, độ đo sự nhất trí của tập thể cho bởi

$\text{PM}(i) = Q_2(\Sigma_j\{\text{IC}(i, j) : j \neq i\} / n-1)$.

2) *Độ trội địa phương.*

Cho A' là tập con của tập phương án A, $A' \subset A$.

Độ trội địa phương là độ đo mức trội hơn của phương án A_i hơn phương án A_j trên tập A' tương ứng với tập con S' của tập nhãn S.

Có thể cố định tập con $S' \subset S$ như sau:

$S' = \{s_{(T+1)/2}, \ldots, s_T\}$

hoặc $S' = \{s_{(T+1)/2+1}, \ldots, s_T\}$.

Định nghĩa 3.3: Sử dụng mảng các mức độ nhất trí $IC(i,j)[s_t]$ (hay độ quan trọng của nhãn) Độ trội địa phương D (*local dominance degree*)của phương án A_i đối với phương án A_j là $D(i,j,S')=\Sigma_t\{IC(i,j)[s_t]: s_t \in S'\}$.

Rõ ràng: $D(i,j,S') \le 1$.

Định nghĩa 3.4: Độ trội địa phương D (*the local Dominance degree of an Alternative*) của phương án A_i trên tập A' tương ứng xét trong S' là tổng các độ trội D của phương án A_i đối với toàn bộ các phương án còn lại của X'.

$D(A_i,A',S')=\Sigma_j\{D(i,j,S'): A_j \in A', j \neq i\} / (|X'|-1)$.

Sự sắp xếp theo quan hệ hơn trong tập S' dựa vào độ trội DA được xác định như sau:

Nếu $D(A_i,A',S') \le D(A_j,A',S')$ với $A_i, A_j \in A'$, thì $A_i \le A_j$.

Như vậy, những phương án trội hơn hẳn của một tập phương án A' sẽ có độ trội D lớn nhất, và có thể gọi là tập trội DS (*The Dominance Set*) của A'.

DS được định nghĩa như sau:

$DS(X') = \{x_i : DA(x_i,X',S') = \max\{DA(x_k,X',S'): x_k \in X'\}\}$

3 Hai quy trình lựa chọn trong bài toán lấy quyết định tập thể

Quy trình 1. Quy trình dựa vào độ nhất trí và toán tử LOWA

Quá trình được thực hiện theo các bước:

a) *Bước* 1. Thu thập các quan hệ mờ $\{P_k\}$. Dựa vào vec tơ trọng số $W=\{w(1), w(2), \ldots, w(k), \ldots, w(m)\}$, chuẩn hoá $w'(k)=w(k)/w_0$, ở đây $w_0=\Sigma_k w(k)$.

b) *Bước* 2. Với mỗi cặp (A_i,A_j) tính trọng số gộp theo từng nhãn s_t đối với mỗi phương án A_i – đó chính là độ nhất trí chọn từ s_t trong khi so sánh cặp của cả Hội đồng.

$$IC(i,j)[s_t] = \Sigma_k\{w'(k): P_k(A_i,A_j)=s_t\}$$

c) *Bước* 3. Chọn hàm lượng hoá Q_1.Tính với mỗi phương án A_i, độ đo sự nhất trí của tập thể cho bởi theo hàm lượng hoá ngôn ngữ Q_2

$PM(i) = Q_2(\Sigma_j\{ IC(i,j): j \neq i \} / n-1)$.

Chọn phương án trích dẫn A_{i*} theo điều kiện

$PM(i^*) = \max\{PM(i) : A_i \in A \}$.

d) *Bước* 4. Dùng toán tử Low tính

$E(i, j) = Low(S, U)$.

với $U=[u_1, \ldots, u_T]$ là vector trọng số, $u_t= IC(i,j)[s_t]$, $t=1, \ldots, T$.

e) *Bước* 5. Dùng $\{E(i,i^*) : A_i \in A\}$ để phân lớp tập phương án A thành các lớp $Y_1, Y_2, \ldots, Y_T$, sao cho:

$Y_t = \{ A_i : E(i,i^*)=s_t \}$, với mỗi $s_t \in S$, $A_{i*} \in Y_{(T+1)/2}$

và lớp Y_{t*} lớn nhất là lớp có:

$t^* = \max\{t : Y_t \neq \emptyset, s_t \in S \}$

f) *Bước* 6. Cố định tập S' và tính toán độ trội địa phương $D(A_i, Y_t, S')$ của mỗi $A_i \in Y_t \neq \emptyset$. Sắp xếp tập phương án Y_t theo độ trội địa phương.

Lời giải của bài toán là $DS(Y_{t*})$.

Quy trình 2. Quy trình dựa nghiệm tập thể mờ FCS và toán tử LOWA

Quá trình được thực hiện theo các bước sau:

a) *Bước* 1. Thu thập các quan hệ mờ $\{P_k\}$. Dựa vào vec tơ trọng số $W=\{w(1), w(2), \ldots, w(k), \ldots, w(m)\}$, chuẩn hoá $w'(k)=w(k)/w_0$, ở đây $w_0=\Sigma_k w(k)$.

b) *Bước* 2. Với mỗi cặp (A_i, A_j) tính trọng số gộp theo từng nhãn s_t đối với mỗi phương án A_i – đó chính là độ nhất trí chọn từ s_t trong khi so sánh cặp của cả Hội đồng.

$IC(i,j)[s_t] = \Sigma_k \{w'(k): P_k(A_i, A_j)=s_t\}$

c) *Bước* 3. Dùng toán tử LOWA tính ma trận quan hệ mờ

$E(i, j) = Low(S, U)$,

với $U=[u_1, \ldots, u_T]$ là vector trọng số, $u_t= IC(i,j)[s_t]$, $t=1, \ldots, T$.

d) *Bước* 4. Tìm nghiệm tập thể mờ FCS

$FCS = (\mu_{FCS}(A_1)/ A_1, \ldots, \mu_{FCS}(A_n)/ A_n)$.

với $\mu_{FCS}(A_i)$ = Low(S,$V(i)$), ở đây vectơ $V(i)=[v_1, \ldots, v_t, \ldots, v_T]$ với mỗi t, $v_t = \#\{j : E(i,j)=s_t, j \neq i\}/m - 1$.

e) *Bước* 5. Phân cụm. Dùng $\{\mu_{FCS}(A_i): A_i \in A\}$ để phân cụm tập phương án A thành các lớp $Y_1, Y_2, \ldots, Y_T$, sao cho:

$Y_t = \{A_i\ \mu_{FCS}(A_i)=s_t\}$, với mỗi $s_t \in S$. Rõ ràng $A=\cup_t Y_t$

Mỗi tập con Y_t có thể sắp xếp theo sự sắp xếp của tập nhãn S:

$Y_t < Y_{t'}$, nếu $t < t'$.

Dùng $\{E(i,i^*) : A_i \in A\}$ để phân lớp tập phương án A thành các lớp $Y_1, Y_2, \ldots, Y_T$, sao cho:

$Y_t = \{A_i : E(i,i^*)=s_t\}$, với mỗi $s_t \in S$, $A_{i^*} \in Y_{(T+1)/2}$

và lớp Y_{t^*} lớn nhất là lớp có:

$t^* = \max\{t : Y_t \neq \emptyset, s_t \in S\}$

f) *Bước* 6. Cố định tập S' và tính toán độ trội địa phương $D(A_i, Y_t, S')$ của mỗi $A_i \in Y_t \neq \emptyset$. Sắp xếp tập phương án Y_t theo độ trội địa phương.

Lời giải của bài toán là DS(Y_{t^*}).

Tài liệu dẫn

[1] **R.R. Yager**: On ordered weighted averaging aggregation operators in multicriteria decision making, IEEE Trans. System Man Cerbernetics. **18** (1988) 183 - 190.

[2] **R.R. Yager**: Families of OWA operators, Fuzzy Sets and Systems, **59** (1993) 125-148.

[3] **D.P.Filev and R.R. Yager**: On the isue of obtaining OWA weights, Fuzzy Sets and Systems, v. 94, (1998), 157-169.

[4] **G. Bordogna, M. Fedrizzi and G. Pasi**: A linguistic modeling of consenssus in group decision making based on OWA operatorrs, IEEE Trans. System Man Cerbernet. 27 (1997), 126 – 132.

[5] **Delgado, J.L. Verdagay and M.A. Vila**: Linguistic decision making models. J. Intelligent Systems **7** (1993) 479-492.

[6] **F.Herrera, E. Herrera Viedma and J.L. Verdegay**: A sequential selection process in group decision making with a linguistic assessement approach, Information Sciences, **85** (1995), 223-239.

[7] **F. Herrera, E. Herrera-Viedma and J.L. Verdegay**: A model of consensus in group decision making under linguistic assessments, Fuzzy Sets and Systems 78 (1996), 73-87.

[8] **F. Herrera, E. Herrera-Viedma and J.L.Verdegay**: Direct approach processes in group decision making using linguistic OWA operators. Fuzzy Sets and Systems 79 (1996) 175 - 190.

[9] **F. Herera, E. Herera-Viedma and J.L. Verdegay**: Choice processes for non-homogeneous group decision making in linguistic setting, Fuzzy Sets and Systems 94 (1998). 287-308.

[10] **B.C.Cuong:** A new process in group decision making using linguistic consensus measures and linguistic OWA operators, Preprint 98/1, Institute of Mathematics, Hanoi, 1998.

[11] **B.C. Cuong:** Some problems in group decision making under linguistic assessments. VJFUZZY'98: Vietnam-Japan Bilateral Symposium on Fuzzy Systems and Applications. Proceeding. Nguyen Hoang Phuong and Ario Ohsato, Eds., Hanoi, 1998.

[12] **B.C.Cuong:** On group decision making under linguistic assessments, Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems. Vol.7, no.4, 1999, 301 – 308.

[13] **B.C.Cuong and Phan V.H.Van:** A choice proces for multiciteria group decision making under linguistic asessments, Proceedings MFI–99, August 26-29, 1999, 403 – 408.

[14] **Bùi Công Cường:** Logic mờ và các ứng dụng đa dạng của nó, Viện Toán học Hà nội, Preprint 99/35, Hà nội 10/ 1999.

[15] **Bùi Công Cường:** Kiến thức cơ sở của hệ mờ, Trường Thu "Hệ mờ và ứng dụng", Lần thứ nhất. Hà nội. 8/ 2000.

[16] **C.L. Hwang and M. J. Lin:** Group Decision Making under Multiple Criteria, Springer-Verlag, Berlin, 1987.

[17] **J. Fodor and M. Roubens:** Fuzzy Preference Modelling and Multicriteria Decision Support, Kluwer Academic Publishers, Dordrecht, 1994.

[18] **L. Zadeh:** Fuzzy sets and information granularity, in M.M.Gupta et al., Eds., Advances in Fuzzy Sets Theory and Applications, North-Holland, London, 1979 3-18.

[19] **L. Zadeh:** A computational approach to fuzzy quantifiers in natural language, Comput. Math. Appls. **9** (1983) 149-184.

11 GIẢI PHÁP DỰ ĐOÁN THÔNG MINH TRONG HỆ HỖ TRỢ QUYẾT ĐỊNH CÁC HOẠT ĐỘNG NGÂN HÀNG

Nguyễn Thanh Thuỷ và Nguyễn Thị Diệu Thư***
** Đại học Bách Khoa Hà nội*
*** Ngân hàng Ngoại Thương Việt Nam*

1 Đặt vấn đề

Vấn đề dự báo vốn đã được quan tâm từ lâu do sự cần thiết của nó. Trong rất nhiều ngành nghề. dự báo đóng một vai trò quan trọng trong việc hoạch định chiến lược phát triển, hay đơn giản hơn. nó được sử dụng thường xuyên công việc hàng ngày như dự báo thời tiết,

Trong hoạt động kinh tế thương mại. lĩnh vực kinh doanh nào cũng tiềm ẩn rủi ro. Song có thể nói Ngân hàng là lĩnh vực kinh doanh tiềm ẩn nhiều rủi ro nhất. Hình thái rủi ro muôn hình. muôn vẻ. Ta có thể kể ra một số loại rủi ro sau đây:

1) *Rủi ro nguồn vốn*:

Ngân hàng là tổ chức trung gian tài chính - đi vay để cho vay. Bởi vậy, Ngân hàng rất dễ có nguy cơ mất khả năng thanh toán khi không đủ vốn và kinh doanh sẽ kém hiệu quả nếu không cho vay được. Khi đó, cho dù các chương trình máy tính quản lý rủi ro trong lĩnh vực nguồn vốn có theo dõi sự biến động hàng ngày về nguồn vốn thì cũng khó đảo ngược được tình thế do trong quá trình giám sát. quản lý nguồn vốn thiếu thông tin về dự đoán nguồn vốn.

2) *Rủi ro tỷ giá hối đoái*:

Vốn kinh doanh của Ngân hàng bao gồm nội tệ và ngoại tệ. Nếu việc sử dụng vốn ngoại tệ lớn và sau đó đồng nội tệ bị lên giá kéo dài thì Ngân hàng sẽ bị thua lỗ. Tình trạng rủi ro về tỷ giá hối đoái còn xảy ra giữa các loại ngoại tệ với nhau khi có biến động. Công nghệ tin học tuy đã được ứng dụng trong Quản lý rủi ro tỷ giá hối đoái nhưng do thiếu thông tin đầu vào nên đầu ra không chính xác. Trong khi đó, kinh nghiệm trong việc theo dõi, dự đoán biến động tỷ giá còn nhiều hạn chế.

3) *Rủi ro trong hoạt động tín dụng và thẩm định đầu tư*:

Đây là loại rủi ro lớn nhất và phức tạp nhất. Tuy nhiên cho đến nay việc ứng dụng công nghệ tin học mới chỉ hạn chế ở các chương trình quản lý tín dụng, cho phép Ngân hàng quản lý các khoản nợ trong hạn và quá hạn của khách hàng một cách tự động. Vì vậy rủi ro tín dụng ở nước ta đang là vấn đề bức thiết cần tìm các giải pháp khắc phục thích hợp. nhất là trong việc hỗ trợ quản lý, dự đoán, cảnh báo các thông tin tín dụng.

2 Giải pháp thông minh xây dựng công cụ dự đoán hỗ trợ việc ra quyết định

Với tình hình thực tế như trên, ta thấy rằng việc xây dựng một công cụ cho phép dự đoán sẽ đem lại nhiều lợi ích cho hoạt động kinh doanh của Ngân hàng cũng như các tổ chức tài chính nói chung. Đã có nhiều giải thuật thống kê được cài đặt tích hợp cũng như được phát triển thành gói công cụ riêng nhằm hỗ trợ cho công tác dự đoán như MS Excel, Lotus, Microfit, v.v. Tuy nhiên, chúng chỉ thực hiện được các chức năng hồi quy đơn giản như hồi quy tuyến tính, hồi quy theo phương pháp bình phương nhỏ nhất, hồi quy phi tuyến bậc thấp, ...Các phương pháp này đều yêu cầu phải biết trước mô hình toán học. Việc dự đoán thực chất chỉ là việc mô hình hoá các hàm tuyến tính hoặc phi tuyến dựa trên các công thức hồi quy, nghĩa là muốn dự đoán được thì phải biết các yếu tố đầu vào phụ thuộc. Điều này không làm được trong việc dự đoán tương lai, khi không biết được các thông tin đầu vào. Ví dụ như trong Excel, hàm dự đoán được xây dựng như sau: FORECAST(giá trị x cần để dự đoán giá trị y, các giá trị y đã biết, các giá trị x đã biết). Hơn nữa, hàm này lại có hạn chế là giá trị của biến dự đoán bị giới hạn chỉ phụ thuộc được vào một biến độc lập, thường thì ở đây là biến thời gian, còn các biến mà nó phụ thuộc thực sự ở đây lại không xét được.

Do vậy, ở đây chúng tôi nghiên cứu một giải pháp mới dựa trên kỹ thuật khai phá dữ liệu nhằm đạt được khả năng và kết quả dự đoán tốt hơn. Kỹ thuật khai phá dữ liệu khá giống với kỹ thuật thống kê theo cách nó xây dựng mô hình dự đoán từ dữ liệu. Tuy nhiên, việc lựa chọn khai phá dữ liệu sẽ có thuận lợi hơn so với phương pháp thống kê truyền thống vì nó đưa ra được cái nhìn tổng thể toàn bộ quá trình hoạt động kinh doanh chứ không chỉ là độ chính xác dự đoán theo thống kê.

Để thực hiện khai phá dữ liệu, quy trình xử lý phải bao gồm: xác định vấn đề cần phải giải quyết, xác định dữ liệu liên quan, thu thập và tiền xử lý dữ liệu sao cho giải thuật khai phá dữ liệu có thể hiểu được, lựa chọn kỹ thuật khai phá dữ liệu và cuối cùng là thực hiện khai phá dữ liệu. Đầu ra của việc khai phá dữ liệu là các mẫu (pattern) mà từ đó, ta rút ra được kết quả dự đoán.

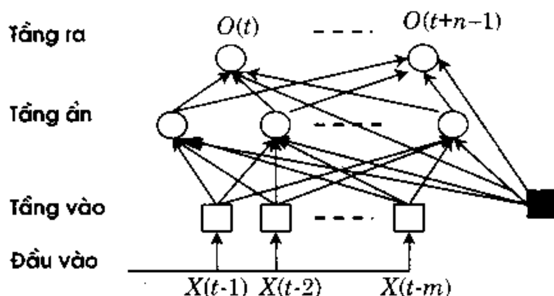
1) Kỹ thuật khai phá dữ liệu

Để giải quyết bài toán dự đoán, kỹ thuật khai phá dữ liệu ta sử dụng ở đây là mạng neuron lan truyền ngược (Back-propagation neural network) kết hợp với giải thuật di truyền (Genetic Algorithm - GA). Mạng neuron là một tiếp cận tính toán mới liên quan đến việc phát triển các cấu trúc toán học với khả năng học. Một trong những ưu điểm phải kể đến của mạng neuron là khả năng tạo ra các mô hình dự đoán có độ chính xác cao, đáp ứng được nhiệm vụ đặt ra của khai phá dữ liệu là dự đoán các sự kiện phụ thuộc vào thời gian. Phương pháp được sử dụng trong mạng neuron lan truyền ngược là phương pháp học có thầy (supervised learning). Hoạt động của nó là phân tích các dữ liệu quá khứ để sinh ra một quy luật dự đoán ra các dữ liệu trong tương lai. Mẫu chiết xuất bằng mạng neuron được thể hiện ở các nút đầu ra của mạng.

Trong mạng lan truyền ngược, mỗi nút được kết hợp với một ngưỡng, vì vậy các mẫu (hay các luật) của một khái niệm là sự kết hợp của các trọng số lớn hơn ngưỡng.

2) **Mạng neuron lan truyền ngược có trễ và cửa sổ dịch chuyển**

Bài toán dự đoán là bài toán sử dụng các dữ liệu mẫu đầu vào thường là dữ liệu theo thời gian và dữ liệu ra cũng là dữ liệu theo thời gian. Việc suy diễn theo thời gian liên quan đến các dữ liệu tại thời điểm hiện tại và các thời điểm trước đó. Mối quan hệ sau trước này là điểm mấu chốt trong việc học các dữ liệu theo thời gian. Thực ra, phần khó nhất trong việc học các mẫu này là làm thế nào để nhận ra, miêu tả và lưu giữ các mối quan hệ đó. Để thực hiện điều đó, ta cải tiến mạng neuron lan truyền ngược thành mạng có trễ và đưa vào khái niệm cửa sổ dịch chuyển.



Hình 1 : Mạng neuron áp dụng cho dữ liệu theo thời gian.

Nếu dữ liệu vào có x bits và được làm trễ với cửa sổ dịch chuyển là m thì sẽ có $m*x$ đơn vị đầu vào để mã hoá mẫu vào. Khi có dữ liệu mới đưa vào, dữ liệu này sẽ được đặt tại nút vào tại một đầu nhất định của mạng, các dữ liệu cũ sẽ được dịch đi một đơn vị trên các nút vào của mạng giống như ở thanh ghi dịch.

Việc dự đoán dựa vào số liệu của bao nhiêu thời điểm thì cửa sổ dịch chuyển đầu vào (lookback windows) là bấy nhiêu. Cũng như vậy, với nhu cầu dự đoán trong những thời điểm xa bao nhiêu thì cửa sổ dịch chuyển đầu ra (look ahead windows) cũng là bấy nhiêu. Một ví dụ về quan hệ vào ra của mạng thực hiện dự đoán với dữ liệu theo thời gian có thể được biểu diễn như sau:

$$Y(t)=f_{nn}(X(t-1),X(t-2),\ \dots\ ,X(t-m))$$

trong đó m là cửa sổ dịch chuyển đầu vào và cũng là số nút vào của mạng, cửa sổ dịch chuyển đầu ra bằng 1 và cũng là số nút ra của mạng. Quan hệ trên cho thấy mạng sẽ dự đoán giá trị Y tại thời điểm t tiếp theo trong tương lai với thời điểm hiện tại là t-1, biết các giá trị của nó và các giá trị phụ thuộc của nó trong quá khứ và hiện tại.

3) **Tích hợp giải thuật di truyền vào mạng neuron**

Với phương pháp tìm kiếm cục bộ để tối thiểu hoá sai số, mạng lan truyền ngược gặp phải một số khó khăn sau:

– Không có khả năng tìm được tập trọng số tối ưu cho một cấu trúc mạng cho trước mà chỉ tìm được một tập trọng số có thể chấp nhận được.

– Mạng có thể không hội tụ hoặc hội tụ sẽ rất chậm làm ảnh hưởng đến tốc độ khai phá dữ liệu cũng như có thể không cho kết quả khai phá dữ liệu như mong muốn.

Trong khi đó, giải thuật di truyền có thể tìm ra vùng quan tâm chứa cực trị toàn cục. Song nó lại không có khả năng đạt được cực trị đó. Thực chất, đây là một giải thuật tối ưu hoá. Chính vì vậy, chúng tôi dùng giải thuật di truyền để hỗ trợ tối ưu hoá cho các thông số của mạng neuron. Chu trình lai ghép được bắt đầu bằng việc khởi tạo một quần thể đầu tiên các nhiễm sắc thể (chính là tập mã hoá các trọng số của mạng) làm đầu vào cho giải thuật di truyền. Giải thuật di truyền sẽ sản sinh ra một thế hệ mới. Thế hệ này được giải mã đưa trở lại mạng neuron để đánh giá độ thích nghi của từng cá thể mà ở đây là các tập trọng số. Trước khi tiếp tục quá trình tiến hoá dựa trên độ thích nghi đã đánh giá được, hệ thống sẽ lưu lại một số cá thể thích nghi nhất của thế hệ hiện tại. Quá trình cứ như thế tiếp tục cho đến khi tìm được một tập trọng số tối ưu. Mạng neuron sau khi đã được tối ưu hoá sẽ đi tìm cực trị cục bộ trong không gian đã được tối ưu hoá này. Sau đó mạng mới được sử dụng để khai phá dữ liệu, chiết xuất ra các mẫu dự đoán.

4) **Mô tả mẫu học**

Khi khai phá dữ liệu, vấn đề chiết xuất dữ liệu và giao diện giữa cơ sở dữ liệu và giải thuật khai phá trở thành vấn đề quan trọng. Thay cho việc đem áp dụng giải thuật khai phá truy nhập vào toàn bộ cơ sở dữ liệu, chúng tôi lựa chọn dữ liệu để phân tích theo yêu cầu của người sử dụng, từ đó dữ liệu sẽ được cấu trúc lại và chuẩn hoá trước khi áp dụng giải thuật. Về lý thuyết thì đây có thể là một vấn đề rất đơn giản, nhưng trên thực tế thì đây quả là một vấn đề khó. Ví dụ như phải lặp đi lặp lại toàn bộ quá trình do dữ liệu thường xuyên thay đổi. Ta thấy rằng sẽ rất cồng kềnh nếu áp dụng giải thuật khai phá dữ liệu với toàn bộ cơ sở dữ liệu. Rất nhiều giải thuật khai phá dữ liệu chỉ dựa trên những thống kê đơn giản, những bảng tóm tắt dữ liệu trong khi đó toàn bộ cơ sở dữ liệu trở nên quá dư thừa, không cần thiết.

Đối với bài toán dự đoán, các mẫu học và mẫu ra có đặc điểm như sau:

– Dữ liệu đầu vào của mạng là một bảng số liệu được lưu trong file dữ liệu (cấu trúc file dạng .xls). Mỗi bộ số liệu nằm trên một hàng (một bản ghi) bao gồm các trường dữ liệu phụ thuộc và dữ liệu độc lập. Có bao nhiêu hàng thì có bấy nhiêu bộ số liệu. Việc số liệu nào được sử dụng làm đầu vào dữ liệu và số liệu nào được sử dụng làm đầu ra đích hoàn toàn do người dùng quyết định. Bảng số liệu này có cấu trúc hoàn toàn giống với một bảng thống kê số liệu bình thường theo thời gian như trong các hoạt động kinh doanh, quản lý, thao tác nghiệp vụ.

– Vậy điều gì ở đây khiến các bảng dữ liệu này trở thành các mẫu học trong dự đoán? Trước hết, ta mô tả một mẫu học của mạng: Đối với bài toán dự đoán, một mẫu học của mạng bao gồm một tập các tín hiệu vào theo thời gian và tín hiệu ra dự đoán mẫu. Tín hiệu ra phụ thuộc vào bao nhiêu thời điểm (lookback windows) thì có bấy nhiêu bộ số liệu vào; tín hiệu ra phụ thuộc vào bao nhiêu tham số độc lập thì một bộ số liệu vào cũng phải có bấy nhiêu tham số. Vì vậy, đầu vào mẫu phải có số tín hiệu bằng tích của

số thời điểm xem xét với số tham số mà nó phụ thuộc. Ví dụ như ta có số cột số liệu vào độc lập trong file dữ liệu là InCol, số thời điểm xem xét là BWnd thì số tín hiệu vào mẫu của mạng là InColxBWnd. Đối với tín hiệu ra, nếu ta cần dự đoán bao nhiêu tham số phụ thuộc thì một bộ số liệu dự đoán phải có bấy nhiêu tham số, mặt khác ta lại muốn dự đoán vào các thời điểm khác nhau trong tương lai nên số tín hiệu đầu ra sẽ là tích của số số liệu phụ thuộc muốn dự đoán với số thời điểm muốn dự đoán. Ví dụ như trong file dữ liệu có OutCol cột số liệu phụ thuộc, nếu muốn dự đoán những thông số này trong vòng thời gian là Awnd thời điểm trong tương lai thì số mẫu đầu ra của mạng phải là OutColxAwnd.

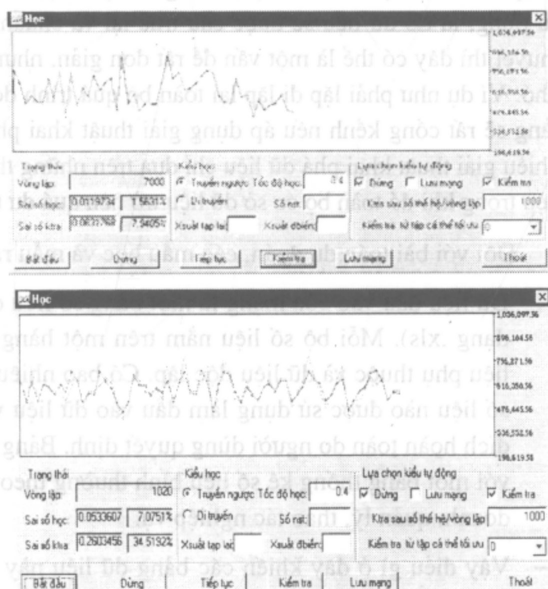
Việc lấy tập dữ liệu mẫu học được thực hiện bằng cách di cửa sổ dịch chuyển đầu vào trên tập mẫu vào và di cửa sổ dịch chuyển đầu ra tương ứng trên tập mẫu ra. Số thứ tự của dữ liệu mẫu đích sẽ được tính bằng số thứ tự bắt đầu của dữ liệu mẫu đầu vào + độ dài của sổ dịch chuyển đầu vào.

Ngoài ra còn có rất nhiều bước quan trọng cần phải làm để tiền xử lý dữ liệu trước khi đưa vào mạng neuron để mạng có thể hiểu được (ví dụ như việc chuẩn hoá dữ liệu, đưa tất cả các tiêu chuẩn dự đoán về dạng số).

3 Kết quả thử nghiệm

Chúng tôi đã thử nghiệm với tập dữ liệu đầu vào là các số liệu doanh thu được theo dõi trong vài năm của một khách hàng xin vay vốn đầu tư tại Ngân hàng (hình 2 trên). Yếu tố đáng quan tâm ở đây là trong những năm tới, liệu tổng doanh thu của khách hàng tăng hay giảm bao nhiêu để có thể xác định nhu cầu vay vốn và khả năng trả nợ của khách hàng.

Để đánh giá hoạt động của việc dự đoán, ta sử dụng một phần của tập dữ liệu để học và các dữ liệu khác dùng để kiểm tra. Đường nét liền trên đồ thị là đường dự đoán. Đường nét đứt là đường giá trị thực (hình 2 dưới).



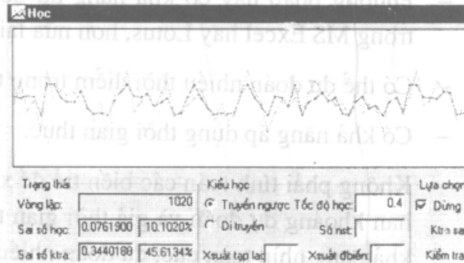
Hình 2: Kết quả thực nghiệm.

Kết quả thử nghiệm cho thấy:

– Với dữ liệu học quá khứ là tổng

doanh thu từ 1994 đến 1996, việc dự đoán doanh thu của tháng tới dựa trên 12 tháng trước đó cho kết quả gần như trùng khít với giá trị thực.

- Với dữ liệu học quá khứ là tổng doanh thu phụ thuộc vào hai yếu tố là lãi thuần và lãi gộp từ 1994 đến 1995, việc dự đoán doanh thu tháng 1 năm 1996 dựa trên 6 tháng trước đó cho kết quả cũng rất khả quan, kết quả dự đoán khá sát với giá trị thực.



- Ta cũng có thể thấy rõ vai trò của giải thuật di truyền trong quá trình học. Hình 3 là kết quả việc học không sử dụng giải thuật di truyền, với số lần học như nhau ta thấy mạng neuron có kết hợp GA cho kết quả tốt hơn.

- Mẫu chiết xuất cũng có thể được thể hiện dưới dạng bảng số liệu. Những số liệu trong 3 hàng cuối là những dữ liệu dự đoán. Ta thấy sai số so với giá trị thực khá nhỏ. Sai số trung bình chỉ khoảng vài phần trăm trong khi kết quả chấp nhận được đối với các phương pháp thống kê hiện nay vào khoảng 10%.

- Bảng mẫu chiết xuất cũng cho thấy kết quả sai số so sánh giữa hai phương pháp có sử dụng GA và không sử dụng GA. Sai số khi không sử dụng GA trong trường hợp này lớn hơn sai số khi có sử dụng GA khoảng 1.5 lần.

	Target	$Output_{GA}$	$Error_{GA}$	$Output_{KGA}$	$Error_{KGA}$
J	581265	562874	18391	524034.8	57230.2
A	579225.1	570850.7	8374.4	558261.3	20963.8
S	645715.6	638964.6	6751	651448.1	-5732.5
O	657985.4	679756.5	-21771.1	659982	-1996.6
N	724053.9	642846.3	81207.6	620352.8	103701.1
D	532932.3	576143.8	-43211.5	627111.8	-94179.5
J-1995	416509.5	459116.5	-42607	514259.7	-97750.2
F	564208.6	557053.1	7155.5	571626.9	-7418.3
M	786707.6	695112.6	91595	704728.3	81979.3
A	597938.5	604482.4	-6543.9	625384	-27445.5
M	648135.9	621468.3	26667.6	593652.1	54483.8
J	518721.4	523443.6	-4722.2	532314.1	-13592.7
J	522032.2	513194.2	8838	514255.3	7776.9
A	572768.2	620891.7	-48123.5	623143.8	-50375.6
S	616184.9	701197.9	-85013	700691.5	-84506.6
O	661245.6	642353.3	18892.3	620478.2	40767.4
N	637526.3	602847.9	34678.4	597040.8	40485.5
D	482233.3	518225.9	-35992.6	520426.6	-38193.3
J-1996	**437362.9**	**516488.6**	**-79125.7**	**515946.3**	**-78583.4**
F	**420269.3**	**562580.1**	**-142311**	**642137.6**	**-221868**
M	**593667.9**	**648910.1**	**-55242.2**	**693267.2**	**-99599.3**
Sai số trung bình			**-92226.2**		**-133350**

Hình 3: Kết quả thực nghiệm.

4 Kết luận

Với việc áp dụng kỹ thuật này, ta thấy việc dự đoán đạt được những ưu điểm hơn hẳn so với việc dùng các phương pháp thống kê truyền thống như sau:

- Không cần biết những thông tin trong tương lai vẫn có thể dự đoán cho một yếu tố mà nó phụ thuộc vào những thông tin ấy.
- Mạng neuron kết hợp với GA trong khai phá dữ liệu có thể tự dự đoán trên chính những số liệu của mình đã có trong quá khứ.
- Phương pháp này có khả năng dự đoán đa biến, không bị hạn chế về số lượng như trong MS Excel hay Lotus, hơn nữa lại cho kết quả dự đoán chính xác hơn.
- Có thể dự đoán nhiều thời điểm trong tương lai.
- Có khả năng áp dụng thời gian thực.
- Không phải tính toán các biến trễ để xác định các biến vào cho dự đoán. Không bị giới hạn khoảng dự đoán và giả thời gian thực như trong các phương pháp thống kê mà có khả năng nhìn toàn cục, dự đoán nhiều thời điểm trong tương lai.
- Không mất nhiều công sức phân tích và tính toán các tham số đầu vào cho một mô hình hồi quy để dự đoán.

Khác với các phương pháp cơ bản là trước khi học cần phải có giai đoạn lấy mẫu để trích chọn các đặc trưng của mẫu; tập mẫu học luôn yêu cầu phải đúng và phải bao được hết các khả năng xảy ra, khai phá dữ liệu như đã cài đặt ở đây không hề sử dụng một biện pháp trích chọn đặc trưng nào. Nó nhận dữ liệu nguyên sơ như đã có trong cơ sở dữ liệu và chiết xuất ra các mẫu dự đoán khá chính xác. Đây chính là đặc trưng cơ bản của khai phá dữ liệu phân biệt nó với các phương pháp khác.

Tài liệu tham khảo

[1] **Alex Berson and Stephen J.Smith:** Data Warehousing, Data Mining, & OLAP. McGraw-Hill 1997.

[2] **Limin Fu:** Neural Networks in Computer Intelligence, 460. McGraw-Hill, Inc. ,1994.

[3] Introduction to Genetic Algorithms Data structures + Genetic Algorithms = Evolution Program. Springer Verlag, 1994

[4] Tạp chí tin học Ngân hàng số 5/1999, 6/1999.

12 ỨNG DỤNG MẠNG NƠRON TRONG TÍNH TOÁN

Đặng Quang Á
Viện Công nghệ Thông tin

1 Mở đầu

Trong vài thập niên qua mạng nơron nhân tạo - một mô hình tính toán nhằm mô phỏng bộ não người đã được sử dụng có hiệu quả trong các lĩnh vực trí tuệ nhân tạo, nhận dạng, xử lý ảnh, xử lý tín hiệu, y học, điều khiển, ... Các loại bài toán chính được giải quyết nhờ mạng nơron là: phân loại, so sánh, tự tổ chức và tối ưu (xem [4, 5, 13]). Trong bài viết này chúng tôi sẽ đề cập đến ứng dụng của mạng nơron trong tính toán, cụ thể là để giải các bài toán tối ưu tổ hợp và hệ phương trình đại số tuyến tính cỡ lớn. Phải nói rằng, đây là cách tiếp cận mới đầy hứa hẹn tới các bài toán đã được giải bằng các phương pháp khác.

Bài viết gồm hai phần. Phần một đề cập việc sử dụng mô hình mạng nơron nhân tạo Hopfield để giải các bài toán tối ưu tổ hợp như bài toán sánh cặp có trọng, bài toán người bán hàng, bài toán bốn mầu, bài toán phân chia đồ thị, ... Phần hai giành cho mô hình mạng nơron với cơ chế phản hồi có khả năng nhớ trạng thái cũ để giải các hệ phương trình đại số tuyến tính thu được sau khi rời rạc hóa các bài toán biên cho phương trình elliptic.

2 Ứng dụng mạng nơron giải các bài toán tối ưu tổ hợp

2.1 Mô hình mạng nơron nhân tạo

Mạng nơron nhân tạo bao gồm hai thành phần: các nút (đơn vị xử lý, nơron) và các liên kết giữa chúng được gán một trọng số nào đó đặc trưng cho cường độ liên kết.

Ta ký hiệu: u_i là tín hiệu đầu vào, V_i là tín hiệu đầu ra của nơron i. Trạng thái đầu vào của nơron i được xác định bởi tổng tuyến tính của các tín hiệu vào có trọng từ các nơron j khác:

$$u_i := \sum_{j \neq i} w_{ij} V_j + \theta_i . \tag{2.1}$$

trong đó θ_i là đầu vào ngoài (external input, offset). Đầu ra của nơron i cho bởi

$$V_i = f_i(u_i), \tag{2.2}$$

trong đó f_i là hàm kích hoạt của nơron i.

Trong đa số các mạng người ta dùng một hàm f chung cho cả mạng. Một số hàm thường được dùng trong thực tế là:

- Hàm McCuloch-Pitts:

$$f(u)=\begin{cases}1 & \text{nếu} \quad u>0 \\ 0 & \text{khác}\end{cases}$$

- Hàm sigmoid:

$$\text{sigmoid}(u)=\frac{1}{1+e^{-2\lambda u}},$$

trong đó λ là hằng số xác định độ nghiêng của hàm.

Sự thay đổi trạng thái đầu vào của các nơron cho bởi phương trình động học

$$\frac{du_i}{dt}=-\frac{\partial E(V_1,V_2,\cdots,V_n)}{\partial V_i} \quad . \tag{2.3}$$

ở đây $E=(V_1,V_2,\ \dots,V_n)$ là hàm năng lượng của mạng gồm n nơron.

Sự hội tụ (hay ổn định) của mạng nơron cho bởi định lý dễ chứng minh sau:

Định lý 1 (xem [9]): Nếu các hàm kích hoạt (1.2) là các hàm không giảm thì $\frac{dE}{dt}\le 0$, tức là năng lượng của mạng sẽ giảm và đạt tới giá trị cực tiểu.

Do đó, nếu phải tìm các gía trị $V_1,V_2,\ \dots,V_n$ để cực tiểu hóa hàm $E=E(V_1,V_2,\ \dots,V_n)$ ta có thể tiến hành theo phương pháp sau: xây dựng mạng nơron (2.3), (2.2) với các hàm kích hoạt không giảm. Khi đó từ một vị trí ban đầu, sau một thời gian mạng sẽ ổn định. Lúc đó đo đầu ra của các nơron ta sẽ được các giá trị $V_1,V_2,\ \dots,V_n$ cần tìm.

2.2 Ánh xạ các bài toán tối ưu tổ hợp lên mạng nơron

Từ sau công trình của Hopfield và Tank (1985) mạng nơron nhân tạo đã được sử dụng nhiều để giải các bài toán tối ưu tổ hợp (xem [5, 6, 9, 12]).

Có thể nêu lên các bước sau đây trong việc sử dụng mạng nơron để giải các bài toán tối ưu hóa hay còn gọi là ánh xạ các bài toán tối ưu lên mạng nơron [6]:

- Chọn sơ đồ biểu diễn để các đầu ra của các nơron có thể giải mã thành các nghiệm có thể của bài toán tối ưu.
- Chọn hàm năng lượng sao cho cực tiểu của nó ứng với nghiệm "tốt nhất" của bài toán cần ánh xạ.

– Gán giá trị cho các tham số của hàm năng lượng - điều này sẽ xác định các trọng số tương đối gán cho các thành phần khác nhau của hàm năng lượng.

– Rút ra phương trình động học của các nơron (tương ứng với việc xác định các trọng số liên kết và đầu vào ngoài).

– Đặt giá trị đầu cho các tín hiệu vào.

Không có phương pháp trực tiếp ánh xạ các bài toán tối ưu có ràng buộc lên mạng nơron ngoại trừ việc thêm vào hàm mục tiêu các thành phần phạt khi các ràng buộc bị phá vỡ. Trong trường hợp hàm năng lượng được biểu diễn như tổng có trọng của hàm mục tiêu của bài toán và các thành phần phạt.

Dưới đây chúng ta sẽ xét một số bài toán cụ thể.

2.2.1 Bài toán sánh cặp có trọng

Phát biểu bài toán: Cho N điểm $i=1,2, \ldots ,N$. Khoảng cách giữa 2 điểm i và j là d_{ij}. Phải nối các điểm thành từng cặp, mỗi điểm chỉ nối với đúng một điểm khác sao cho tổng độ dài của các đường nối là nhỏ nhất.

Hình 1 dưới đây chỉ ra một thí dụ khi $N=6$.



Hình 1 : Lời giải cho N=6.

(a) Lời giải tốt (b) Lời giải xấu

Ánh xạ bài toán lên mạng nơron:

Ta gán cho mỗi cặp điểm (i,j) với $i<j$ một nơron ij với đầu ra là n_{ij}. Như vậy cần tất cả $N(N-1)/2$ nơron. Mỗi phương án của bài toán ứng với một trạng thái của mạng với

$$n_{ij} = \begin{cases} 1 & \text{nếu các điểm } i \text{ và } j \text{ có nối nhau} \\ 0 & \text{nếu khác} \end{cases}$$

Ta cần phải giải bài toán

$$L = \sum_{i<j} d_{ij} n_{ij} \to \min$$

với các ràng buộc $\sum_{j} n_{ij} = 1$, $\forall\ i=1,2, \ldots ,N$.

Các ràng buộc trên nói lên rằng mỗi điểm phải được nối với đúng một điểm khác.

Ta định nghĩa $n_{ij} = n_{ji}$ khi $j<i$ và $n_{ii}=0$. Đưa thành phần phạt vào hàm cần cực tiểu hóa ta đi đến hàm năng lượng

$$E[n] = \frac{1}{2}\sum_{i,j} d_{ij}n_{ij} + \frac{\gamma}{2}\sum_{i}(1-\sum_{j} n_{ij})^2 \ ,$$

trong đó γ là tham số. Từ đây lấy đạo hàm riêng $\partial E/\partial n_{ij}$ ta xác định được phương trình động học của các nơron

$$\frac{du_{ij}}{dt} = \gamma \sum_{k} n_{ik} - n_{ij} + \gamma.$$

Nếu ký hiệu cường độ liên kết giữa nơron ij với nơron kl là $w_{ij,kl}$ và đầu vào ngoài của nơron ij là θ_{ij} ta có

$$w_{ij,kl} = \begin{cases} 1 & \text{nếu } kl \text{ có một chỉ số chung với } ij \\ 0 & \text{nếu khác} \end{cases}$$

$$\theta_{ij} = d_{ij} - \gamma.$$

Chọn hàm kích hoạt của các nơron là hàm McCulloch-Pitts. Và như vậy là ta đã ánh xạ được bài toán sánh cặp có trọng lên mạng nơron nhân tạo.

2.2.2 Bài toán người bán hàng

Bài toán này tương tự bài toán sánh cặp có trọng nhưng khó giải trên máy tính hơn vì nó là bài toán NP-đầy đủ (xem [11]).

Phát biểu bài toán: Cho N điểm (thành phố) với các khoảng cách là d_{ij}. Tìm đường đi khép kín ngắn nhất sao cho mỗi điểm chỉ đi qua một lần và trở về điểm xuất phát.

Đã có nhiều tài liệu đề cập bài toán này do các ứng dụng thực tiễn phong phú của nó. Cách tiếp cận nơron tới bài toán này do Hopfield và Tank đề xuất năm 1985-1986.

Ánh xạ bài toán lên mạng nơron:

Ta dùng các biến nhị phân n_{ia} để biểu diễn các nghiệm có thể

$$n_{ia} = \begin{cases} 1 & \text{nếu thành phố } i \text{ là điểm dừng thứ } a \text{ của lộ trình} \\ 0 & \text{nếu khác} \end{cases}$$

Ta cần phải giải bài toán tìm lộ trình ngắn nhất.

$$L = \frac{1}{2}\sum_{ij,a} d_{ij}n_{ia}(n_{j,a-1} + n_{j,a-1}) \rightarrow \min$$

với các ràng buộc

$$\sum_a n_{ia} = 1 \text{ , } \forall\ i=1,2,\ \ldots\ ,N \text{ (đối với mỗi thành phố)}$$

$$\sum_i n_{ia} = 1 \text{ , } \forall\ a=1,2,\ \ldots\ ,N \text{ (đối với mỗi điểm dừng)}$$

Thêm các thành phần phạt vào hàm độ dài lộ trình ta đi đến hàm cần cực tiểu hóa

$$E = \frac{1}{2}\sum_{ij,a} d_{ij} n_{ia}(n_{j,a+1} + n_{j,a-1}) + \frac{\gamma}{2}\left[\sum_a\left(1-\sum_i n_{ia}\right)^2 + \sum_i\left(1-\sum_a n_{ia}\right)^2\right] \tag{2.4}$$

trong đó γ là tham số dương.

Bây giờ đối với mỗi điểm dừng a ta dùng N nơron để biểu diễn và cũng ký hiệu n_{ia} là đầu ra của nơron thứ i của điểm dừng a. Như vậy ta cần $N \times N$ nơron để biểu diễn N điểm dừng của lộ trình. Thí dụ về biểu diễn nơron của bài toán với $N=4$ cho bởi hình 2.

Điểm dừng

1 2 3 4

Thành phố 1

2

3

4

Hình 2: Thí dụ về biểu diễn nơron của bài toán với N=4.

Lấy đạo hàm riêng của hàm năng lượng (2.4) và đổi dấu ta xác định được phương trình động học của các nơron. Sau đó chọn hàm kích hoạt của các nơron, ta xây dựng được mạng nơron nhân tạo cho bài toán người bán hàng.

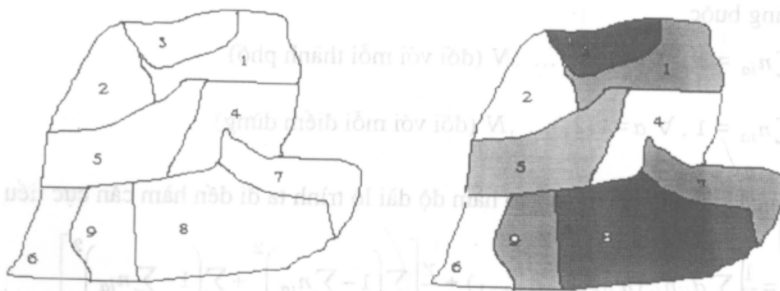
2.2.3 Bài toán bốn mầu

Phát biểu bài toán: Cho một bản đồ trong đó có N nước (miền). Cần tô mầu sao cho hai nước kề nhau tức là có chung biên giới được tô bởi hai mầu khác nhau.

Bài toán này được dễ dàng giải quyết nếu sử dụng số lượng lớn các mầu. Nhưng bài toán sẽ khó hơn rất nhiều nếu phải dùng số mầu nhỏ nhất để tô mầu một bản đồ cho trước. Nếu số nước nhỏ thì bài toán cũng sẽ dễ hơn. Các nhà toán học đã chứng minh rằng bản đồ bất kỳ có thể tô bởi 4 mầu.

Ánh xạ bài toán lên mạng nơron: 4 mầu được biểu diễn bởi 4 nơron. Mỗi nước cần 4 nơron để được gán một mầu.

Thí dụ bản đồ 9 nước được tô bởi 4 mầu như trong hình 3.



Hình 3 : Bản đồ 9 nước và 4 mầu

Biểu diễn nơron cho bài toán tô mầu 9 nước cho bởi hình 4, trong đó sử dụng mảng 4×9 nơron.

Đỏ
Vàng
Xanh
Lá cây

	1	2	3	4	5	6	7	8	9
1					■				
2		■		■		■			
3			■					■	
4	■						■		■

Hình 4 : Biểu diễn nơron cho bản đồ 4 mầu trong hình 3

Để đặc trưng cho sự kề cận của 2 nước X và Y ta dùng biến d_{XY} được xác định như sau:

$$d_{XY} = \begin{cases} 1 & \text{nếu } X \text{ và } Y \text{ kề nhau} \\ 0 & \text{nếu khác} \end{cases}$$

Ký hiệu V_{XY} là đầu ra của nơron thứ i trong nước X:

$$V_{Xi} = \begin{cases} 1 & \text{nếu } X \text{ tô bằng mầu } i \\ 0 & \text{nếu khác} \end{cases}$$

Bài toán đặt ra là phải cực tiểu hoá hàm

$$\sum_{i=1}^{4} \sum_{X=1}^{N} \sum_{\substack{Y=1 \\ Y \neq X}}^{N} d_{XY} V_{Xi} V_{Yi}$$

với ràng buộc $\sum_{i=1}^{4} V_{Xi} = 1$, $\forall\ X = 1, 2, \ \ldots\ , N$ (mỗi nước phải tô một mầu).

Đưa thành phần phạt vào hàm mục tiêu khi ràng buộc bị phá vỡ ta đi đến hàm được coi là hàm năng lượng

$$E = \sum_{X=1}^{N} \sum_{\substack{Y=1 \\ Y \neq X}}^{N} d_{XY} V_{Xi} V_{Yi} + \frac{\gamma}{2} \sum_{X=1}^{N} \left(\sum_{i=1}^{4} V_{Xi} - 1 \right)^2 . \tag{2.5}$$

Lấy đạo hàm riêng của E theo V_{Xi} và đổi dấu ta được phương trình động học của các nơron

$$\frac{du_{Xi}}{dt} = -\sum_{\substack{Y=1\\Y\neq X}}^{N} d_{XY}V_{Yi} - \gamma\left(\sum_{j=1}^{1} V_{Xj} - 1\right).$$

2.2.4 Một số bài toán khác

Cách tiếp cận nơron đã được sử dụng để giải nhiều bài toán tối ưu tổ hợp quan trọng khác như bài toán phân chia đồ thị (graph partioning), bài toán E–hậu (E–queen), bài toán định tuyến (channel routing), bài toán sắp xếp và tìm kiếm, bài toán khôi phục ảnh do nhiễu hoặc mờ ....

Có thể tìm hiểu kỹ về các bài toán này trong bài tổng quan [6] và các sách [4, 9].

2.3 Tìm trạng thái ổn định của mạng

Sau khi các bài toán tối ưu đã được ánh xạ lên mạng nơron nhân tạo tức là ta đã xây dựng được mạng nơron, công việc tiếp theo là phải tìm trạng thái ổn định của mạng. Để làm việc này ta phải giải số phương trình động học của mạng và xác định đầu ra của các nơron tại mỗi bước thời gian.

Giả sử ta có mạng $N \times N$ nơron với đầu vào là u_{ij} và đầu ra là v_{ij} với hàm kích hoạt McCuloch-Pitts. Khi đó thuật toán tìm trạng thái ổn định của mạng có thể mô phỏng bởi chương trình trong mã giả sau:

```
Begin
      Khởi tạo các giá trị đầu của u_ij và v_ij
      While (điều kiện xác định trạng thái ổn định chưa thoả mãn) do
      Begin
      /*** Vòng lặp 1
            for i:=1 to N
            for j:=1 to to N
            u_ij = u_ij+Δu_ij
      /***Kết thúc vòng lặp 1
      /*** Vòng lặp 2
            for i:=1 to N
            for j:=1 to to N
            if u_ij > 0 then v_ij = 1 else v_ij = 0
      /***Kết thúc vòng lặp 2
      end;
end;
```

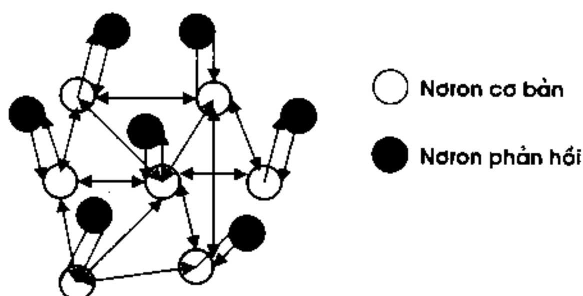
Trong thuật toán trên công thức $u_{ij} = u_{ij} + \Delta u_{ij}$ chính là phương pháp Euler áp dụng cho phương trình động học.

Chú ý rằng trọng số của các thành phần phạt xác định độ nhạy tương đối tới năng lượng của các ràng buộc. Nó đóng vai trò rất quan trọng trong sự hội tụ của mạng. Việc chọn trọng số thích hợp phụ thuộc vào bài toán cụ thể.

3 Ứng dụng mạng nơron giải hệ phương trình đại số tuyến tính

3.1 Mạng nơron với cơ chế phản hồi

Để ý rằng trong quy tắc chuyển trạng thái cho bởi phương trình (1.1), (1.2) đầu ra của nơron i được xác định bởi đầu ra của các nơron khác nối với nơron i, chứ giá trị của chính nơron i không được sử dụng một cách truy toán trong quá trình chuyển trạng thái. Điều đó có nghĩa là mạng không nhớ hay không phản hồi trạng thái trước của nó. Dưới đây là một cấu hình cải biên của mạng nơron được đề xuất trong [10] để thực hiện cơ chế phản hồi.



Hình 5 : Mạng nơron với cơ chế phản hồi

Một khái niệm nơron mới được đưa vào mạng. Để tránh nhầm lẫn ta sẽ gọi các nơron không có cơ chế phản hồi là các nơron cơ bản (base neuron), các nơron có cơ chế phản hồi là các nơron phản hồi (feedback neuron). Hình 5 chỉ ra cấu trúc của mạng với cơ chế phản hồi.

Trong mô hình này mỗi nơron cơ bản có một nơron phản hồi tương ứng. Do đó, số nơron phản hồi bằng số nơron cơ bản. Hoạt động của nơron phản hồi cũng được mô tả bởi (1.1), (1.2) ngoại trừ một điều là trọng giữa các nơron phản hồi và đối tác của nó là không đối xứng, nghĩa là nếu nơron cơ bản i có nơron phản hồi là i' thì $w_{ii'} \neq w_{i'i}$.

Quy tắc chuyển trạng thái của mạng phản hồi như sau:

Đối với nơron cơ bản:

$$t_i := \sum_{j \neq i} w_{ij} u_j + w_{ii} w_{i'} + \theta_i \, . \tag{3.1}$$

$$u_i = f_i(t_i). \tag{3.2}$$

Đối với nơron phản hồi:

$$t_{i'} := w_{i'i} u_i + \theta_{i'}$$ (3.3)

$$u_{i'} = f_{i'}(t_{i'}).$$ (3.4)

trong đó: i' chỉ nơron phản hồi của nơron cơ bản i, u_j và $u_{i'}$ trong (3.4) là đầu vào từ các nơron j và i' tới nơron i. u_i trong (3.2) là đầu ra từ nơron i, w_{ij} là trọng đối xứng, $w_{ii'}$ là trọng xác định từ nơron i' tới nơron i, trong khi đó $w_{i'i}$ xác định trong hướng ngược lại, θ_i và $\theta_{i'}$ các đầu vào từ ngoài, $t_{i'}$ và $t_{i'}$ là các biến trạng thái, f_i và $f_{i'}$ là các hàm kích hoạt. (chú ý sự thay đổi ký hiệu so với phần 1).

Năng lượng của mạng được xác định như sau

$$E(u) = -\frac{1}{2}\sum_{\substack{i,j\\ i\neq j}} w_{ij}u_i - \sum_i \theta_i u_i - \frac{1}{2}\sum_i w_{ii'}u_i u_{i'} - \frac{1}{2}\sum_{i'} w_{i'i}u_{i'}u_i - \sum_{i'}\theta_i u_{i'}$$ (3.5)

Mạng nơron với cơ chế phản hồi mô tả ở trên sẽ được sử dụng để giải các hệ phương trình đại số tuyến tính thu được sau khi rời rạc hóa các bài toán biên cho phương trình elliptic bằng phương pháp phần tử hữu hạn hoặc phương pháp sai phân.

3.2 Nhắc qua về một số phương pháp lặp giải hệ phương trình đại số tuyến tính

Sau khi rời rạc hóa các bài toán biên đối với phương trình elliptic người ta thu được hệ phương trình đại số tuyến tính

$$Au = b$$ (3.6)

trong đó $A = (a_{ij})$ là ma trận đối xứng xác định dương cỡ $n \times n$, b là véc tơ n chiều, n bằng số nút lưới của miền được xét. Nghiệm của (3.6) cho nghiệm của bài toán cực tiểu hóa phiếm hàm

$$J(u) = \frac{1}{2}(Au, u) - (b, u),$$ (3.7)

ở đây, như thường lệ, $(\cdot,\cdot)$ ký hiệu tích vô hướng trong R^n.

Có nhiều phương pháp trực tiếp và phương pháp lặp giải hệ (3.7) (xem, chẳng hạn [2, 3, 7, 8]). Dưới đây chúng ta nhắc lại một số phương pháp lặp quen biết giải (2.6), mà có thể được xem như các quy tắc chuyển trạng thái của mạng nơron sẽ được xây dựng.

Phương pháp Jacobi: Xuất phát từ $u(0) \in R^n$, tính $u(t), t = 1, 2, \ldots$ sử dụng phép lặp

$$u_i(t+1) = u_i(t) - \frac{1}{a_{ii}} g_i(t),$$ (3.8)

trong đó $g_i(t)=\sum_j a_{ij}u_i(t)-b_i$. (3.9)

1) Phương pháp Gauss-Seidel:

$$u_i(t+1)=u_i(t)-\frac{1}{a_{ii}}\left(\sum_{j<i}a_{ij}u_j(t+1)+\sum_{j\geq i}a_{ij}u_j(t)-b_i\right) \quad (3.10)$$

2) Phương pháp giảm dư trên (SOR):

$$u_i(t+1)=u_i(t)-\frac{\omega}{a_{ii}}\left(\sum_{j<i}a_{ij}u_j(t+1)+\sum_{j\geq i}a_{ij}u_j(t)-b_i\right) \quad (3.11)$$

3) Phương pháp lặp đơn giản:

$$u_i(t+1)=u_i(t)-\tau g_i(t). \quad (3.12)$$

4) Phương pháp lặp Chebyshev:

$$u_i(t+1)=u_i(t)-\tau_t g_i(t),\ t=0,1,\ldots,T-1. \quad (3.13)$$

trong đó $\{\tau_t\}$ là bộ tham số Chebyshev ([6, 7]).

3.3 Các thuật toán nơron

3.3.1 Các thuật toán nơron tương ứng các phương pháp Jacobi, Gauss-Seidel và SOR

Để xây dựng thuật toán nơron giải hệ (2.6) hay bài toán tối ưu (2.7) ta chỉ ra các trọng, các đầu vào và các hàm kích hoạt của mạng nơron với cơ chế phản hồi đã nêu ở mục 2.1.

Đối với nơron cơ bản:

$$w_{ij}=w_{ji}=-\omega a_{ij}. \quad (3.14)$$

$$\theta_{ij}=\omega b_i. \quad (3.15)$$

$$f_i(t_i)=f_i(t_i)/a_{ii}. \quad (3.16)$$

Đối với nơron phản hồi

$$w_{ii'}=(1-\omega)a_{ii}. \quad (3.17)$$

$$w_{i'i}=-a_{ii}. \quad (3.18)$$

$$\theta_{i'}=0 \quad (3.19)$$

$$f_{i'}(t_{i'})=-t_{i'}/a_{ii}. \quad (3.20)$$

trong đó ω tham số dương điều khiển biên độ phản hồi để tang tốc quá trình cực tiểu hóa.

Từ (3.3), (3.4) và (3.18) ÷ (3.20) rút ra được $u_{i'}=u_i$, nghĩa là đầu ra từ nơron cơ bản được sao chép và lưu giữ như đầu ra của nơron phản hồi.

Giả sử $u_i(t)$ và $E(t)$ là đầu ra và năng lượng tại bước t. Ta viết lại quy tắc chuyển trạng thái (2.1), (2.2) trong dạng

$$u_i(t+1)=u_i(t) - \frac{\omega}{a_{ii}} g_i(t) \tag{3.21}$$

trong đó ký hiệu

$$g_i(t)= \sum_j a_{ij}u_t(t) - b_i. \tag{3.22}$$

Xét một số trường hợp khác nhau của (3.21).

1) *Trường hợp* $\omega = 1$ (không phản hồi): Ta có

$$u_i(t+1)=u_t(t) - \frac{1}{a_{ii}} g_i(t). \tag{3.23}$$

Nếu tại mỗi bước thời gian t tất cả các nơron được kích hoạt, tức là trạng thái của tất cả các nơron được cập nhật đồng bộ thì quy tắc chuyển trạng thái (3.23) chính là công thức của phương pháp Jacobi (2.8). Nếu khác, tức là nếu các nơron được kích hoạt tuần tự sao cho tại mỗi bước thời gian chỉ có một nơron được kích hoạt, trong khi các nơron khác giữ nguyên không đổi thì quy tắc chuyển trạng thái là

$$u_l(t+1)=u_l(t) - \frac{1}{a_{ll}} g_l(t). \tag{3.24a}$$

$$u_i(t+1)=u_t(t), \qquad i \neq l. \tag{3.24b}$$

trong đó l là nơron duy nhất được kích hoạt tại bước t.

Dễ thấy rằng (3.24) chính là công thức (2.10) của phương pháp Gauss-Seidel.

2) Trường hợp $\omega \neq 1$ (có phản hồi): nếu các nơron được kích hoạt không đồng bộ như mô tả ở trên thì quy tắc chuyển trạng thái là

$$u_l(t+1)=u_l(t) - \frac{\omega}{a_{ll}} g_l(t).$$

$$u_i(t+1)=u_i(t), \qquad i \neq l.$$

Đây chính là phương pháp SOR (3.11).

Sự hội tụ của mạng với các quy tắc chuyển trạng thái nêu trên đã được thiết lập trong [1, 10].

3.3.2 Các thuật toán nơron tương ứng các phương pháp lặp đơn giản và lặp Chebyshev

Bây giờ ta chọn các trọng, các đầu vào ngoài và các hàm kích hoạt của mạng nơron với cơ chế phản hồi như sau

1) *Đối với các nơron cơ bản*:

$$w_{ij} = w_{ji} = -\tau a_{ij} .$$

$$\theta_i = \tau b_i .$$

$$f_i(t_i) = t_i .$$

2) *Đối với các nơron phản hồi*:

$$w_{ii'} = (1-\tau)a_{ii} .$$

$$w_{i'i} = -1$$

$$\theta_{i'} = 0$$

$$f_{i'}(t_{i'}) = -t_{i'} .$$

Đối với mạng này ta cũng có $u_{i'} = u_i$ và năng lượng của mạng là $E(u) = \tau J(u)$.

Quy tắc chuyển trạng thái đối với nơron cơ bản của mạng là

$$u_i(t+1) = u_i(t) - \tau g_i(t).$$

Đây chính là công thức của phương pháp lặp đơn giản (2.12). Trong [1] đã thiết lập được điều kiện hội tụ của mạng là $I > \frac{\tau}{2} A$.

Bây giờ giả sử γ_1 và γ_2 là các cận dưới và trên, tương ứng của ma trận A, nghiã là $\gamma_1 I \leq A \leq \gamma_2 I$ và $\{\tau_t\ ,\ t=0,1,\dots\ ,T-1\}$ là bộ số Chebyshev xây dựng theo các cận γ_1 và γ_2 (chi tiết xem [7, 8]). Ta chọn các trọng, các đầu vào ngoài và các hàm kích hoạt của mạng nơron với cơ chế phản hồi như sau

1) *Đối với các nơron cơ bản*:

$$w_{ij} = w_{ij}(t) = -\tau_t a_{ij} .$$

$$\theta_{ij} = \theta_{ij}(t) = \tau_t b_i .$$

$$f_i(t_i) = t_i .$$

2) *Đối với các nơron phản hồi*:

$$w_{ii'} = w_{ii'}(t) = 1 - \tau_t a_{ii} .$$

$$w_{i'i} = -1$$

$$\theta_{i'} = 0$$

$$f_{i'}(t_{i'}) = -t_{i'} .$$

Rõ ràng là trong mạng này các trọng, các đầu vào ngoài thay đổi trong quá trình cực tiểu hóa. Quy tắc chuyển trạng thái đối với nơron cơ bản là:

$$u_i(t+1)=u_i(t)-\tau_t g_i(t).$$

Đây chính là công thức (2.13) của phương pháp Chebyshev đối với hệ (3.6).

Trong [1] đã thực hiện việc phân tích so sánh tốc độ hội tụ của các thuật toán nơron nêu trên.

Tài liệu tham khảo

[1] **Dang Quang A:** On neural network approach to the solution of boundary value problems for elliptic equations, in: VJFUZZY'98, Vietnam-Japan bilateral symposium on fuzzy systemsd applications (Halong Bay, 30/9-2/7), Proceedings, Hà nội, 1998, 415-420.

[2] **D. P. Bertsekas, J. N. Tsitsiklis:** Parallel and Distributed Computation, Numerical Methods, Prentice - Hall, 1988.

[3] **L. A. Hageman, D. M. Young:** Applied iterative Methods, Akademic Press, 1981.

[4] **J. Hertz, A. Krogh, R. G. Palmer:** Introduction to the Theory of Neural Computation, Addison-Wesley, 1991.

[5] **LiMin Fu:** Neural Network in Computer intelligence, Mc. Graw Hill, 1994.

[6] **J. Ramanujam and P. Sadayappan:** Mapping Combinatorial Optimization Problems onto Neural Networks, information Sciences. 82. 239-255 (1995).

[7] **A. A. Samarskij:** Introduction to Numerical Methods. Moscow. Nauka. 1987 (in Russian).

[8] **Samarskij & E. Nikolaev:** Numerical Methods for Grid Equations, V. 2, Birkhauser Basel. 1989.

[9] **Y. Takefuji:** Neural Network Parallel Computing, Kluwer Acad. Publ., 1992.

[10] **G. Yagawa and H. Okuda:** Finite element solutions with feedback network mechanism through direct minimization of energy functionals, int. J. Numer. Meth. Engng, **39**, 867-883 (1996).

[11] **H. Papadimitriou & K. Kenneth:** Combinatorial optimization, Prentice-Hall, 1982 (bản dịch tiếng Nga, "Mir", 1985).

[12] **C. Pain, C. R. De Oliveira and A. J. Goddard:** A Neural network graph partioning procedure for grid-based domain decomposition, Int. J. Numer. Meth. Engng, 44, 593-613 (1999).

[13] **R. J. Schalkoff:** Artificial neural networks, McGraw-Hill, International Editions, 1997.

13 MỘT SỐ VẤN ĐỀ NHẬN DẠNG MÔ HÌNH VÀ ĐIỀU KHIỂN SỬ DỤNG MẠNG NƠRON

Vũ Như Lân
Viện Công nghệ Thông tin

I Nhận dạng phi tuyến mô hình hệ động lực

1.1 Nhận dạng thông số hệ thống (off line)

Trong quá trình điều khiển các đối tượng động lực, cần phải giải quyết bài toán nhận dạng thông số mô hình hệ động lực. Hiện nay có hai hướng cơ bản mô tả toán học các đối tượng động lực:

- Mô hình hàm truyền
- Mô hình không gian trạng thái

Loại mô hình hàm truyền phù hợp với giai đoạn đầu phát triển lý thuyết điều khiển và hướng đến các hệ tuyến tính dừng.

Loại mô hình không gian trạng thái tổng quát hơn và có thể hướng đến lớp đối tượng rộng hơn như hệ phi tuyến, dừng và không dừng.

Quan điểm không gian trạng thái tỏ ra rất hiệu quả trong các nghiên cứu khoa học và trong thiết kế các hệ động lực phức tạp.

Mục tiêu của bài toán nhận dạng không ngoài việc đảm bảo cho hiệu quả điều khiển.Tuy nhiên bài toán nhận dạng có thể có ý nghĩa độc lập. Trong trường hợp này đòi hỏi độ chính xác của các ước lượng thông số nhận được. Đây là bài toán nhận dạng off-line mô hình với cấu trúc cho trước.

Bài toán nhận dạng thông số off-line:

Quan sát được các vectơ $z(t)$ bao gồm vectơ trạng thái với nhiễu tác động $v(t)$ và đầu vào $u(t)$ như sau:

$$Z(t)=h\left[x(t),u(t),v(t),P_2(t),t\right], \tag{1.1.1}$$

ở đây $P_2(t)$ là các thông số chưa biết của hệ thống.

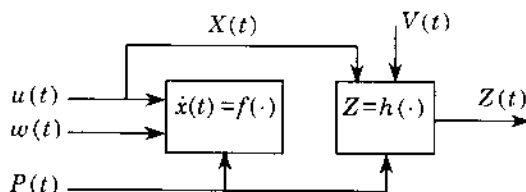
Vectơ trạng thái của hệ được mô tả bởi phương trình:

$$\dot{x}(t) = f\left[x(t),u(t),w(t),P_1(t),t\right], \tag{1.1.2}$$

trong đó $w(t)$ là vectơ nhiễu tác động từ bên ngoài.

Cần tìm các thông số mô hình đảm bảo cực trị một tiêu chuẩn nhận dạng.

Sơ đồ tổng quát có dạng biểu diễn ở hình 1



Hình 1: Sơ đồ tổng quát nhận dạng thông số mô hình

Véc tơ thông số $P(t)=[P_1(t), P_2(t)]$ có thể chứa các hệ số của phương trình vi phân, phương trình quan sát và đồng thời có thể có các đặc trưng thống kê của nhiễu $v(t)$, $w(t)$.

1.1.1 Phương pháp xấp xỉ vi phân

Nếu lấy vi phân giá trị các biến tại các thời điểm, thì có thể xây dựng hệ phương trình tuyến tính được giải bằng phương pháp bình phương cực tiểu đối với véc tơ cần tìm P. Nếu $x(t), \dot{x}(t)$, $u(t)$ là các hàm đã biết thì phương trình (1.1.2) có thể viết dưới dạng:

$$\begin{pmatrix} \dot{x}(t_1) \\ \vdots \\ \dot{x}(t_k) \end{pmatrix} = \begin{pmatrix} \text{Ma trận } A \text{ chứa các hàm} \\ \text{phi tuyến } x \text{ và } u \text{ tại} \\ t_1, t_2, \dots, t_k \end{pmatrix} \begin{pmatrix} p_1 \\ \vdots \\ p_m \end{pmatrix} \qquad (1.1.3)$$

trong đó $\dot{x}(t_i)$ là ước lượng của $x(t_i)$ được tính theo phương trình mô hình.

Phương pháp bình phương cực tiểu cho kết quả sau:

$$\hat{P}_1 = [A^T A]^{-1} A^T \dot{x}(t). \qquad (1.1.4)$$

Xấp xỉ vi phân đơn giản nhưng không tiện lợi ở mấy điểm sau đây:

- Phải có đạo hàm của $x(t)$ theo thời gian,
- Khi có nhiều tác động thì kết quả nhận được là xấp xỉ trung bình bình phương đến $\dot{x}(t)$ mà không phải $x(t)$.
- Khi không đo được toàn bộ vectơ trạng thái thì phương pháp trên không dùng được.

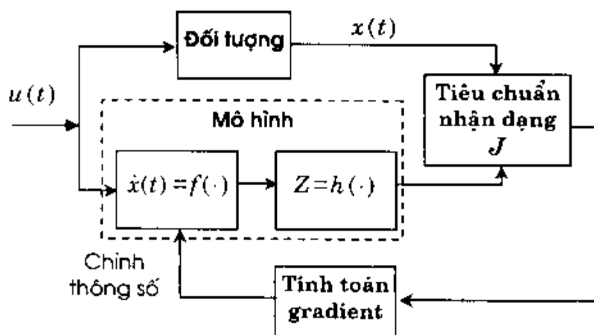
1.1.2 Phương pháp gradient

Giả thiết rằng mô hình phi tuyến (1.1.1) và (1.1.2) được biểu diễn dưới dạng rời rạc. Cần xác định véc tơ thông số P sao cho $x(t)$ với độ chính xác cho trước phù hợp với $z(t)$ dưới tác động của điều khiển $u(t)$.

So sánh $x(t)$ với $z(t)$ có thể dẫn đến tiêu chuẩn sai số J bao gồm hiệu các đầu ra của mô hình và đối tượng (hệ thống):

$$J = \sum_{i=0}^{k} H\left[x(t_i) - z(t_i)\right] \tag{1.1.5}$$

Trong đó H là hàm và thường được chọn dưới dạng tổng bình phương các thành phần vectơ sai số. Cấu trúc hệ nhận dạng theo phương pháp gradient được thể hiện ở hình 2.



Hình 2: Nhận dạng theo phương pháp gradient

Thuật toán nhận dạng gradient như sau:

1) Cho các giá trị ban đầu P_0.
2) Giải các phương trình sai phân hoặc vi phân và xác định được J.
3) Cho $p_i = p_{i0} + \Delta$ và giải cũng các phương trình đó, xác định được $\partial J / \partial p_i$.
4) Thông tin nhận được về hướng gradient được sử dụng tuỳ theo từng trường hợp để xây dựng thuật toán tìm vectơ thông số P.

Thuật toán gradient lặp đơn giản nhất để xác định thông số P, là phương pháp hạ nhanh nhất. Hướng của phương pháp hạ nhanh nhất ngược với hướng gradient và ở điểm ban đầu trùng với hướng trong đó tiêu chuẩn sai số giảm nhanh nhất. Có nghĩa là hướng của phương pháp hạ nhanh nhất được mô tả bằng vectơ:

$$P^{(k+1)} = P^{(k)} + \Delta P, \tag{1.1.6}$$

$$\Delta P = [\Delta p_1, \Delta p_2, \ldots, \Delta p_m,]^T.$$

trong đó:

$$\Delta p_i = -C \frac{\partial J}{\partial p_i} \bigg/ \left[\sum_{j=1}^{m} \left(\frac{\partial J}{\partial p_j}\right)^2 \right]^{\frac{1}{2}} \tag{1.1.7}$$

Lưu ý rằng $\frac{\partial J}{\partial p_j}$ thường được xấp xỉ như sau:

$$\frac{\partial J}{\partial p_j}=\frac{J(p_1,p_2,\ldots,p_j+\Delta,\ldots,p_m)-J(p_1,p_2,\ldots,p_j,\ldots,p_m)}{\Delta} \tag{1.1.8}$$

Hằng số C của phương trình (1.1.7) xác định bước thay đổi vectơ thông số theo hướng gradient. Nếu cho C quá lớn thì tiêu chuẩn sai số nhận dạng J thực tế có thể cũng rất lớn. Ngược lại, chọn C quá nhỏ thì tốc độ hội tụ có thể quá chậm. Vì vậy cần chọn $C = C^*$ tối ưu theo nghĩa cực tiểu J theo hướng ngược với gradent

$$J(P + C^*\Delta P) = \min_C [\, J(P + C\Delta P)].$$

Để tìm C^* có thể sử dụng các phương pháp tối ưu thông thường [38].

1.1.3 Phương pháp tìm kiếm trực tiếp

Phương pháp này không yêu cầu biết trước giá trị đạo hàm (sai phân) như các phương pháp gradient và xấp xỉ đạo hàm. Mặc dù phương pháp tìm kiếm hội tụ chậm hơn so với các phương pháp khác nhưng trên thực tế được sử dụng khá nhiều do tính đơn giản, dễ sử dụng của nó.

Bản chất của phương pháp dựa trên giả thiết rằng độ lệch của vectơ thông số ở những bước tìm kiếm đúng đắn trước đó có thể dẫn đến thành công ở bước sau.

Đầu tiên chọn giá trị ban đầu của vectơ thông số và tính toán hàm mục tiêu tìm kiếm $J(0)$. Sau đó tiến hành xem xét (với bước tính toán cho trước) các hướng phù hợp với tất cả các thành phần của vectơ thông số. Nếu $J(k)<J(0)$ thì chọn lại giá trị ban đầu mới và dịch chuyển "sơ đồ" tính toán sang toạ độ gốc mới và lặp lại chu trình tìm kiếm cho tới khi tìm được giá trị cực tiểu J^*.

$$p_{im}^{(k+1)} = p_i^{(k+1)} + \alpha[\, p_i^{(k-1)} - p_{ic}^{(k)}\,] \tag{1.1.9}$$

trong đó $p_{im}^{(k+1)}$, $p_{ic}^{(k+1)}$ là các tọa độ gốc mới và cũ, $\alpha \geq 1$ là hệ số khuyếch đại.

Tiêu biểu cho trường phái nhận dạng thông số này là Rastrigin L.A[41]. Thuật toán di truyền là thế hệ thuật toán phát triển cao trên ý tưởng tìm kiếm ngẫu nhiên.

1.1.4 Phương pháp tựa tuyến tính

Phương pháp tựa tuyến tính kết hợp với phương pháp bình phương cực tiểu có thể nhận dạng vectơ thông số chính xác hơn khi biết các giá trị xấp xỉ của nó.

Giả sử hệ được mô tả bằng phương trình sau:

$$\dot{x}(t) = f[x,u,P,t],\;\; x(0)=x_0. \tag{1.1.10}$$

Nếu tuyến tính hoá vế phải biểu thức (1.1.10) qua chuỗi Taylor thì có thể tìm P hết sức đơn giản bằng phương pháp bình phương cực tiểu như đã giới thiệu ở trên. Tuy nhiên cần bổ sung một hệ phương trình đánh giá thông số cho (1.1.10) như sau:

$$\dot{p}_i = 0; \qquad p_i(0) = p_{i0}; \quad i = 1, 2, \ldots, m.$$

Như vậy mô hình đánh giá (1.1.10) được mở rộng với:

$$\begin{aligned}
x^T &= [x_1, x_2, \ldots, x_v, p_1, p_2, \ldots, p_m] \\
U^T &= [u_1, u_2, \ldots, u_v, 0, \ldots, 0] \\
f^T &= [f_1(x,u,t), f_2(x,u,t), \ldots, f_v(x,u,t), 0, \ldots, 0] \\
x_0 &= [x_{10}, x_{20}, \ldots, x_{v0}, p_{10}, p_{20}, \ldots, p_{m0}]
\end{aligned} \qquad (1.1.12)$$

Lưu ý rằng có thể dùng phương pháp xấp xỉ vi phân ở những bước đầu tiên của thuật toán tựa tuyến tính.

1.1. 5 Phương pháp sử dụng hàm nhạy

Đây là phương pháp trực giác cho phép xác định thông số tương đối chính xác. Giả sử hệ có dạng (1.1.10). Hàm ma trận nhạy của đầu ra hệ thống được xác định bằng:

$$\lambda = \frac{\partial x}{\partial p} \qquad (1.1.13)$$

hoặc

$$\frac{\Delta x_i}{x_i} \approx \lambda_{ij} \frac{\Delta p_j}{p_j}$$

Sử dụng (1.10) và (1.13) có thể viết

$$\frac{\partial^2 x}{\partial p \partial t} = \frac{\partial f}{\partial x} \cdot \frac{\partial x}{\partial p} + \frac{\partial f}{\partial p} \qquad (1.1.14)$$

$$\dot{\lambda} = \frac{\partial f}{\partial x} \lambda^T + \frac{\partial f}{\partial p}, \qquad \lambda(0) = \frac{\partial x_0}{\partial p}. \qquad (1.1.15)$$

Lấy tích phân (1.1.15) nhận được λ phục vụ cho quá trình nhận dạng.

1.2 Nhận dạng thông số hệ thống (ON-LINE)

Các phương pháp nhận dạng OFF-LINE có nhược điểm chung sau đây:

- Mất thông tin do phép rời rạc hoá.
- Khó thể hiện bằng phần cứng trên thực tế,

– Khi số thông số lớn (>3), khó xác định chính xác vectơ thông số.

– Không sử dụng được khi hệ không dừng.

Mục tiêu của nhận dạng là phục vụ cho bài toán điều khiển. Vì vậy các quan sát thu được theo thời gian phải sử dụng để nhận dạng mô hình tốt hơn đồng thời hiệu chỉnh điều khiển. Như vậy nhận dạng và thiết kế hệ điều khiển phải được tiến hành đồng thời.

Trong chế độ on-line, mô hình phải thật đơn giản. Số các thông số chọn đủ nhỏ và cấu trúc mô hình tuyến tính theo thông số.

Thuật toán nhận dạng on-line được xây dựng sao cho trên mỗi bước tính không cần xử lý lại toàn bộ chuỗi quan sát, có nghĩa là sử dụng quá trình lặp. Sau đây là một số phương pháp nhận dạng thông số hệ thống on-line.

1.2.1 Phương pháp lặp bình phương cực tiểu

Hệ thống có thể mô tả bằng phương trình sai phân tuyến tính theo thông số hoặc điều khiển sau đây:

$$x(k+1)=\Phi(k)P(k)+w(k), \tag{1.2.1}$$

$$z(k)=x(k)+v(k), \tag{1.2.2}$$

trong đó $\Phi(k)=\Phi(x,u,k)$.

Sơ đồ nhận dạng có tính đến hệ số trọng cho các quan sát trong quá khứ theo luật hàm exponent:

$$\hat{P}(k)=\hat{P}(k+1)+K(k)\left[x(k-1)-\Phi(k)\,\hat{P}(k-1)\right], \tag{1.2.3}$$

$$K(k)=P(k-1)\Phi^T(k)\left[\Phi(k)P(k-1)\Phi^T(k)+e^{\Delta T/\tau}\right]^{-1}. \tag{1.2.4}$$

$$P(k)=e^{\Delta T/\tau}\left[I-K(k)\Phi(k)\right]P(k-1). \tag{1.2.5}$$

trong đó ΔT là khoảng cách giữa hai quan sát, τ là thời gian đặc trưng cho khoảng ảnh hưởng tiếp tục của quan sát lên quá trình ước lượng.

1.2.2 Phương pháp xấp xỉ ngẫu nhiên

Thuật toán có dạng sau:

$$\hat{P}(k+1)=\hat{P}(k)+0.5\rho(k)\Delta_p J. \tag{1.2.6}$$

trong đó $\rho(k)$ là vectơ thông số hiệu chỉnh thoả mãn các điều kiện sau:

$$\rho(k)\geq 0, \qquad \sum_{k=0}^{\infty}\rho(k)=\infty, \quad \sum_{k=0}^{\infty}\rho^2(k)<\infty,$$

$$J=e^2(k+1),$$

$$e(k+1)=x(k+1)-\Phi(k+1)\,\hat{P}(k).$$

Như vậy (1.2.6) có thể viết dưới dạng:

$$\hat{P}(k+1) = \hat{P}(k)+\rho(k)\Phi(k)\left[x(k+1)-\Phi(k+1)\,\hat{P}(k)\right]. \tag{1.2.7}$$

Thuật toán xấp xỉ ngẫu nhiên đơn giản hơn thuật toán lặp bình phương cực tiểu, tuy nhiên kém chính xác hơn.

1.2.3 Phương pháp lọc Kalman mở rộng

Lọc Kalman là thuật toán xử lý thông tin sử dụng đầy đủ thông tin tiên nghiệm (cấu trúc, thông số, các đặc trưng thống kê của nhiễu trạng thái và nhiễu quan sát, các dữ kiện về điều kiện ban đầu ...). Nếu trạng thái hoá vectơ thông số $P(k+1)=P(k)$, ta có vectơ trạng thái mở rộng

$$y(k+1)= \left[x(k+1)\ ,P(k+1)\right]^T.$$

và như vậy bộ lọc Kalman mở rộng có thể được sử dụng để xác định đồng thời trạng thái và thông số.

Giả sử hệ có động học:

$$x(k+1)=\Phi(k)\left[x(k),u(k),P_1(k),k\right]+w(k), \tag{1.2.8}$$

$$z(k)=h\left[x(k),u(k),P_2(k),k\right]+v(k), \tag{1.2.9}$$

trong đó:

$$E\{w(j)\}=0\ ;\ E\{v(j)\}=0 \tag{1.2.10}$$

$$\operatorname{cov}\{w(k),w(j)\}=v_k(k)\delta(k-j), \tag{1.2.11}$$

$$\operatorname{cov}\{v(k),v(j)\}=v_v(k)\delta(k-j). \tag{1.2.12}$$

Nếu biết cấu trúc Φ, h và các thông số mô hình P_1, P_2 thì bộ lọc Kalman cho kết quả lọc:

$$\hat{x}(k+1)=\hat{x}(k+1/k)+\chi(k+1)\left\{z(k+1)-h\left[\hat{x}(k+1/k),u(k+1),P_2(k),k+1\right]\right\}. \tag{1.2.13}$$

trong đó dự báo

$$\hat{x}(k+1/k)=\Phi(k)\left[\hat{x}(k),u(k),P_1(k),k\right]. \tag{1.2.14}$$

Ma trận hiệp phương sai của sai số dự báo thoả mãn phương trình:

$$V_x(k+1/k)=\frac{\partial\Phi\left[\hat{x}(k),u(k),P_1(k),k\right]}{\partial\hat{x}(k)}V_x(k)\,\frac{\partial\Phi^T\left[\hat{x}(k),u(k),P_1(k),k\right]}{\partial\hat{x}(k)}+V_w(k) \quad (1.2.15)$$

Ma trận hiệp phương sai của sai số lọc thoả mãn phương trình:

$$V_x(k+1)=V_x(k+1/k)-V_x(k+1/k)\,\frac{\partial h^T\left[u(k+1),\hat{x}(k+1/k),P_2(k),k+1\right]}{\partial\hat{x}(k+1/k)}\times$$

$$\left[\frac{\partial h\left[u(k+1),\hat{x}(k+1/k),P_2(k),k+1\right]}{\partial\hat{x}(k+1/k)}V_x(k+1/k)\times\right. \quad (1.2.16)$$

$$\left.\times\frac{\partial h^T\left[\hat{x}(k+1/k),u(k+1),P_2(k),k+1\right]}{\partial\hat{x}(k+1/k)}+V_x(k+1)\right]^{-1}\times$$

$$\times\frac{\partial h\left[\hat{x}(k+1/k),u(k+1),P_2(k),k+1\right]}{\partial\hat{x}(k+1/k)}V_x(k+1/k).$$

Hệ số Kalman được tính bằng biểu thức sau:

$$K(k+1)=V_x(k+1)\,\frac{\partial h\left[\hat{x}(k+1/k),u(k+1),P_2(k),k+1\right]}{\partial\hat{x}(k+1/k)}V_x^{-1}(k+1) \quad (1.2.17)$$

Các điều kiện ban đầu:

$$\hat{x}=E\{x_0\} \text{ và } V_x(0)=V_{x(0)}. \quad (1.2.18)$$

Do các vectơ thông số $P_1(k)$, $P_2(k)$ thay đổi theo thời gian chưa biết trước nên cần thiết nhận dạng thông số cùng với trạng thái. Tuy nhiên phải giả thiết rằng $P_1(k)$ và $P_2(k)$ trong khoảng thời gian đủ ngắn là không đổi (có nghĩa là đối tượng gần dừng). Khi đó vectơ mở rộng có thể viết dưới dạng sau:

$$y(k+1)=\begin{pmatrix}x(k+1)\\P_1(k+1)\\P_2(k+1)\end{pmatrix}=\begin{pmatrix}\Phi\left[x(k),u(k),k\right]\\P_1(k)\\P_2(k)\end{pmatrix}+\begin{pmatrix}w(k)\\0\\0\end{pmatrix}. \quad (1.2.19)$$

Sử dụng thuật toán (1.2.13) ÷ (1.2.18) đánh giá đồng thời thông số và trạng thái hệ thống với vectơ trạng thái mở rộng (1.2.19).

Lưu ý rằng phương pháp trên chỉ có hiệu quả khi tính phi tuyến thấp.

1.3 Kết luận

Trên đây đã tóm tắt một số phương pháp nhận dạng phi tuyến đơn giản. Kết quả đạt được của các phương pháp trên đã được sử dụng trên thực tế nhưng hạn chế ở các đối tượng

có tính phi tuyến thấp. Khi đối tượng có tính phi tuyến cao, độ bất định lớn và số chiều lớn thì cần phải có tiếp cận khác. Các phần tiếp theo sẽ giới thiệu về mạng nơron nhân tạo và một hướng giải quyết bài toán nhận dạng mô hình hướng đến điều khiển thông minh trên cơ sở sử dụng khả năng học của mạng nơron nhân tạo.

2 Nhận dạng mô hình và điều khiển sử dụng mạng nơron

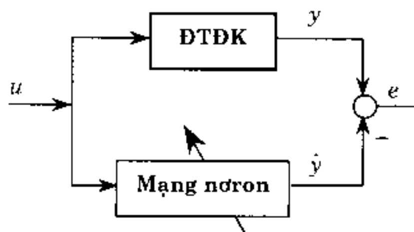
2.1 Mở đầu

Chúng ta sẽ làm việc với mạng nơron nhân tạo. Kiến thức cơ sở về mạng nơron nhân tạo bạn đọc có thể tham khảo bài báo [????, trong sách này trang 3-]. Các bài toán quan trọng trong lý thuyết điều khiển thường là các thuật toán điều khiển và các phương pháp nhận dạng mô hình. Đối tượng điều khiển (ĐTĐK) nói chung là phi tuyến, có độ phức tạp cao, độ bất định lớn. Sự hiểu biết về mô hình đối tượng có thể nghèo nàn do hạn chế về tri thức. Vì vậy mối liên hệ ngược là đặc điểm điển hình được sử dụng trong hệ điều khiển nhằm làm giảm độ bất định của đối tượng và môi trường, đạt đến tính ổn định bền vững. Tuy nhiên khi độ bất định quá lớn, bộ điều khiển không còn phù hợp. Trong trường hợp này cần đến điều khiển thích nghi. Ở đây các thông số của đối tượng được nhận dạng on-line và thông tin này được sử dụng để thay đổi tham số của bộ điều khiển. Tất cả thực hiện được nhờ mạng nơron. Mạng nơron phản hồi rất quan trọng trong hệ điều khiển đối tượng có động học phức tạp. Đối với mạng nơron truyền thẳng nhiều lớp, một tính chất trung tâm cho các ứng dụng của điều khiển là tính xấp xỉ hàm. Các mạng như vậy tạo ra những biến đổi vào-ra, có thể xấp xỉ mọi hàm liên tục với độ chính xác cho trước. Nền tảng cho tính xấp xỉ hàm của mạng nơron nhiều lớp là định lý Kolmogorov [18] và các định lý Stone-Weierstrass [36].

Mạng nơron có thể kết hợp cho cả hai bài toán nhận dạng mô hình và điều khiển [16]. Tuy nhiên trong các hệ điều khiển sử dụng mạng nơron chưa chú ý đúng mức đến tính ổn định của hệ thống.

2.2 Nhận dạng thông số sử dụng mạng nơron

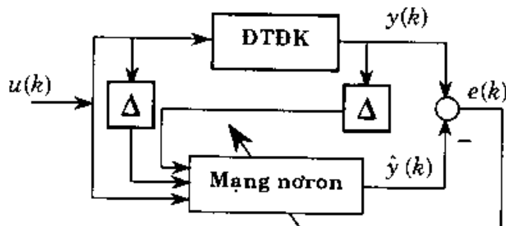
Nhận dạng thông số chính là luyện mạng. Mô hình cơ bản của mạng nơron được luyện để mô phỏng hành vi của đối tượng điều khiển giống như mô hình truyền thống được biểu diễn ở hình 3.



Hình 3: Mô hình nhận dạng cơ bản

Tín hiệu sai số $e=y-\hat{y}$ là cơ sở cho quá trình luyện mạng. Mạng nơron ở đây có thể là mạng truyền thẳng nhiều lớp hoặc các dạng khác. Thuật luyện mạng cũng rất đa dạng. Có thể sử dụng tất cả các thuật nhận dạng đã có [23], các thuật luyện có giám sát [29].
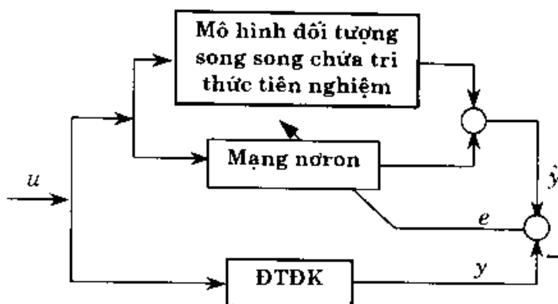
Dạng thông tin vào mạng nơron có thể bổ sung, ví dụ hình 4



Hình 4: Bổ sung thông tin đầu vào cho mạng nơron

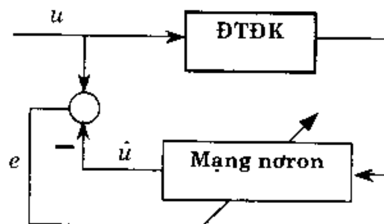
Δ – thời gian trễ.

Lưu ý rằng số lượng các trễ đầu vào và đầu ra của ĐTĐK cần chọn ít nhất cũng phải bằng bậc của ĐTĐK. Nếu có tri thức tiên nghiệm về ĐTĐK thì có thể nhúng trong mô hình của đối tượng dưới dạng mô hình song song như trong [1]. Thông tin của tri thức tiên nghiệm được thể hiện bằng mô hình đối tượng song song thể hiện qua hình 5.



Hình 5: Sử dụng tri thức tiên nghiệm

Ngoài ra có thể thay vì luyện mạng để nhận dạng động học thuận của đối tượng, người ta có thể luyện mạng để nhận dạng động học nghịch như hình 6.



Hình 6: Nhận dạng động học nghịch

Kiểu nhận dạng này phù hợp với những phương pháp điều khiển tiền định. Dạng thông tin được sử dụng trong mạng nơron nhận dạng động học nghịch của đối tượng cũng có thể rất đa dạng tương tự như trường hợp nhận dạng động học thuận hình 4.

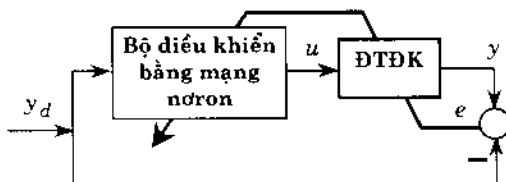
Cần lưu ý rằng phải đảm bảo điều kiện tồn tại động học nghịch của ĐTĐK.

2.3 Điều khiển sử dụng mạng nơron

Mạng nơron có thể được sử dụng như bộ điều khiển truyền thống trong điều khiển theo vòng hở hoặc điều khiển theo vòng kín (điều khiển có phản hồi).Quá trình luyện mạng có thể cài trong luật điều khiển thích nghi. Riêng bài toán điều khiển trong thời gian thực, tốc độ hội tụ của thuật luyện mạng đóng vai trò quan trọng nhất. Nhìn chung, các ứng dụng cho thấy: ở bài toán điều khiển theo vòng hở, đối tượng là ổn định, còn bất định nằm ở thông số đối tượng. Các tác động bên ngoài được loại trừ hoặc có thể bỏ qua. Trong tình huống này mạng nơron phải xấp xỉ được động học nghịch của đối tượng nhằm đạt được tình huống điều khiển lý tưởng. Có khá nhiều ứng dụng của phương pháp trên, đặc biệt cho robot [40].

2.3.1 Điều khiển theo vòng hở

Thay cho việc luyện mạng để mô hình hoá động học nghịch của ĐTĐK, mạng nơron có thể được luyện trực tiếp như một bộ điều khiển theo vòng hở như hình 7 dưới đây:



Hình 7 : Bộ điều khiển thể hiện bằng mạng nơron trong cấu trúc điều khiển theo vòng hở.

Sai số $e = y - y_d$ được sử dụng để luyện mạng. Trong mô hình này cho trước đầu ra mong muốn y_d. Vì vậy thông tin về sai số phải lan truyền ngược qua cả đối tượng điều khiển và mạng nơron để hiệu chỉnh lại thông số mạng. Lưu ý rằng tình huống này luôn xuất hiện trong điều khiển cả theo vòng kín và vòng hở. Như vậy học có giám sát không được sử dụng ở đây. Đầu ra mong muốn của mạng không biết trước nhưng phải được xác định để sử dụng thuật lan truyền ngược sai số. Khi sử dụng phương pháp này để luyện mạng có thể coi ĐTĐK như "lớp đầu ra " của mạng nơron. Nhưng cũng có thể tránh quá trình lan truyền ngược qua đối tượng bằng cách sử dụng thêm mô hình mạng nơron của ĐTĐK. Mô hình mạng nơron này nhận được sau khi nhận dạng mô hình đối tượng. Như vậy sai số có thể dễ dàng lan truyền ngược qua mô hình mạng của ĐTĐK. Nếu sử dụng được mô hình mạng nơron mô phỏng động học nghịch của đối tượng thì mô hình mạng này cho phép lan truyền trực tiếp sai số sang mô hình mạng nơron của bộ điều khiển.

2.3.2 Điều khiển theo vòng kín

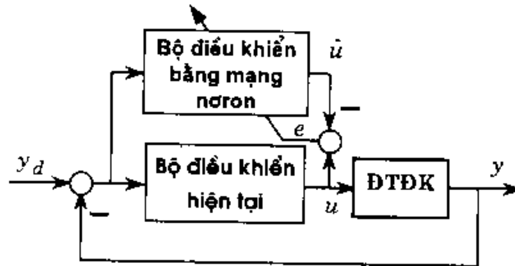
Mạng nơron trong cấu trúc điều khiển theo vòng kín đóng vai trò bộ điều khiển với phản hồi như hình 8.

Nên lưu ý trường hợp này, đầu ra mong muốn của mạng nơron u thể hiện bộ điều khiển phải được xác định từ đầu ra mong muốn của ĐTĐK y_d trước khi sử dụng mọi thuật học có giám sát như lan truyền ngược sai số.



Hình 8 : Bộ điều khiển thể hiện bằng mạng nơron trong cấu trúc điều khiển theo vòng kín.

Có thể có quan điểm khác về bộ điều khiển bằng mạng nơron là mạng nơron được luyện để "bắt chước" bộ điều khiển hiện tại như ở hình 9. Trường hợp này hay xảy ra khi bộ điều khiển hiện tại đang sử dụng quá đắt hoặc không tin cậy. Sau khi bộ điều khiển bằng mạng nơron thay thế bộ điều khiển hiện tại, nó có thể được hiệu chỉnh qua quá trình luyện mạng bám theo sự biến đổi động học của đối tượng và môi trường.



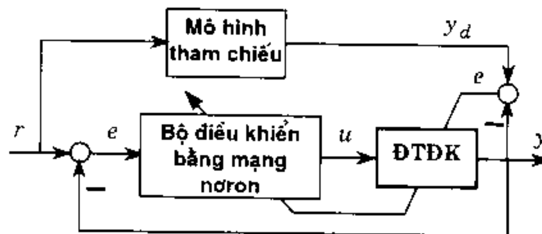
Hình 9 : Mạng nơron được luyện bắt chước bộ điều khiển.

Việc sử dụng mạng nơron như trên tương đương với việc thiết kế và sử dụng một hệ chuyên gia. Cần thận trọng với tiệm cận này vì phải rất linh hoạt. Có thể thay thế cho bộ điều khiển hiện tại theo từng giai đoạn sao cho hợp lý và kinh tế.

2.3.3 Điều khiển với mô hình tham chiếu

Mạng nơron thay thế bộ điều khiển cũng có thể được luyện để làm giảm sai số giữa đầu ra của đối tượng điều khiển và mô hình tham chiếu như ở hình 10.

Ở đây một lần nữa đầu ra mong muốn của bộ điều khiển bằng mạng nơron u phải được xác định trước khi sử dụng thuật lan truyền ngược sai số. Sai số $e=y_d-y$ được truyền qua ĐTĐK và sau đó được sử dụng để chỉnh trọng số liên kết của bộ điều khiển bằng mạng nơron. Tuy nhiên sai



Hình 10 : Điều khiển với mô hình tham chiếu và sai số lan truyền qua ĐTĐK.

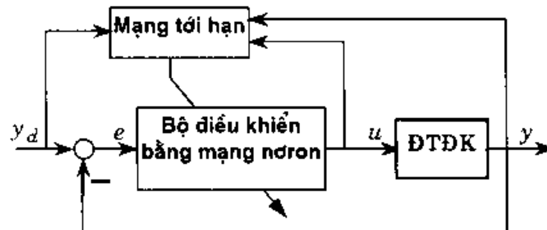
số e cũng có thể không lan truyền qua ĐTĐK mà qua mô hình mạng nơron mô phỏng ĐTĐK hoặc mô phỏng động học nghịch của ĐTĐK tương tự như phần điều khiển theo vòng hở đã trình bày ở trên.

2.3.4 Điều khiển theo thời gian vượt quá (over time)

Trong các bài toán điều khiển sử dụng mạng nơron xem xét ở các phần trước, quỹ đạo mong muốn của đầu ra ĐTĐK thường được biết trước. Điều đó có nghĩa rằng đầu ra mong muốn của mạng nơron có thể được biết trước hoặc có thể tính được trước. Bài toán điển hình của dạng này là bài toán điều chỉnh và bài toán bám quỹ đạo [15,16]. Khi quỹ đạo mong muốn của đối tượng điều khiển không được biết trước thì thuật học có giám sát không sử dụng được. Một trong những bài toán không biết trước quỹ đạo mong muốn của ĐTĐK là điều khiển làm cực tiểu năng lượng hoặc thời gian trong quá trình đạt đến trạng thái cần thiết nào đó của ĐTĐK. Ví dụ bài toán cực tiểu thời gian vượt quá là yêu cầu và hiệu quả của các tác động hiện tại lên các kết quả của tương lai. Bài toán này có nhiều trong điều khiển robot thông minh hoặc xử lý ô nhiễm sao cho ảnh hưởng về ô nhiễm môi trường của tác nhân ô nhiễm nào đó lên môi trường xung quanh sẽ chấm dứt trong thời gian ngắn nhất. Hiện tại có hai phương pháp được sử dụng:

1) Xây dựng mô hình của quá trình và sau đó sử dụng một dạng nào đó của thủ tục lan truyền ngược theo thời gian như có ở [17, 25]. Phương pháp này khó sử dụng dưới dạng tổng quát.
2) Sử dụng giá trị tới hạn thích nghi (adaptive critic) và các phương pháp học củng cố. Phương pháp này bao gồm hai mạng như hình 11.

Bộ điều khiển bằng mạng nơron là mạng hoạt động chính, còn mạng tới hạn có thể là mạng nơron hoặc không. Mạng tới hạn sẽ xấp xỉ lời giải quy hoạch động cho bài toán điều khiển tối ưu năng lượng hoặc thời gian vượt quá. Mạng này phải đảm bảo hoạt động tốt trong môi trường có nhiễu hoặc với mô hình không chính xác của ĐTĐK [37]. Mạng tới hạn phải tạo ra chỉ số thực hiện J theo ý đồ người thiết kế và chỉ dẫn cho bộ điều khiển cần thích nghi như thế nào theo chỉ số thực hiện đó.



Hình 11 : Điều khiển bằng mạng nơron sử dụng mạng tới hạn.

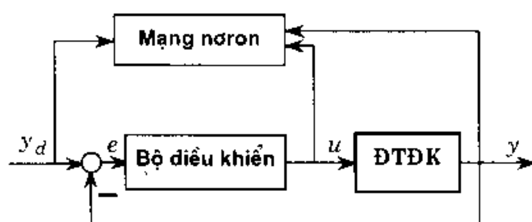
Khi điều khiển u làm cho chỉ số J tăng lên ở bước tiếp theo, bộ điều khiển bằng nơron sẽ hoạt động theo thuật toán "thưởng". Khi điều khiển u làm cho chỉ số J giảm đi ở bước sau đó thì bộ điều khiển bằng mạng nơron hoạt động theo thuật toán "phạt". Trong thuật học không có giám sát, không có thông tin về đầu ra mong muốn và hệ học chỉ nhận được đánh giá về việc thực hiện quá trình học đó theo từng bước. Lưu ý rằng học có giám sát nếu

có thể được sử dụng sẽ tốt hơn học củng cố. Để thiết kế mạng nơron tới hạn phải xác định được kết quả thực hiện hiện tại của ĐTĐK để "thưởng" hoặc "phạt" bộ điều khiển bằng nơron. Như vậy, mạng tới hạn thích nghi và các phương pháp học củng cố là cách hợp lý để giải quyết bài toán điều khiển tối ưu thời gian vượt quá.

2.3.5 Bộ điều khiển với hỗ trợ quyết định của mạng nơron

Khi bộ tới hạn thích nghi là mạng nơron có thể xây dựng được bộ điều khiển thông minh mức thấp. Trong đó mạng nơron đóng vai trò bộ lịch trình (scheduler) [33] sẽ quyết định luật điều khiển nào được sử dụng. Xem hình 12.

Mạng nơron cũng có thể được luyện để xác định giá trị của các thông số trong bộ điều khiển công nghiệp PID thông thường [2]. Ngoài ra mạng nơron có thể đóng vai trò bộ tối ưu tìm giá trị tối ưu của hàm mục tiêu điều khiển [28]. Đầu ra của mạng nơron là giá trị thông số của bộ điều khiển làm cực tiểu hàm giá. Mạng nơron còn cung cấp những thông tin làm việc sai lệch cho bộ điều khiển, giúp bộ điều khiển hoạt động chính xác [1].



Hình 12: Điều khiển với bộ lịch trình.

2.4 Kết luận

Mạng nơron là hệ xử lý thông tin thế hệ mới, đặc biệt có ý nghĩa quan trọng trong bài toán nhận dạng mô hình và điều khiển hệ phi tuyến phức tạp, chứa bất định cao. Ngày nay nhiều kết quả thu được hứa hẹn một tiềm năng ứng dụng lớn của mạng nơron trong điều khiển và mô hình hoá hệ thống. Tuy nhiên nhiều vấn đề còn mở, đòi hỏi những kết quả lý thuyết mạnh hơn và những ứng dụng toàn diện hơn trong đời sống.

3 Nhận dạng mô hình và điều khiển sử dụng mạng nơron đối xứng xuyên tâm cơ sở

3.1 Hàm đối xứng xuyên tâm cơ sở và ứng dụng trong nhận dạng

Hàm đối xứng xuyên tâm cơ sở (Radial Basis Functions) viết ngắn là RBF đã có từ lâu trong lý thuyết xấp xỉ và được sử dụng để xấp xỉ hàm chưa biết dựa trên cơ sở các cặp điểm vào - ra biểu diễn hàm chưa biết đó. Như vậy RBF có thể xem như một dạng mạng nơron.

Trong nhận dạng mô hình hệ thống RBF có thể biểu diễn theo cấu trúc mạng perceptron. Mọi hệ phi tuyến có thể xấp xỉ bằng RBF. Đây là đặc điểm làm cho RBF đặc biệt phù hợp với bài toán nhận dạng mô hình. Về nguyên tắc RBF có thể được coi là mạng

nơron 3 lớp (tương tự mạng perceptron với 1 lớp ẩn). Tuy nhiên RBF có đặc điểm sau: Đầu ra luôn là biến đổi tuyến tính đối với trọng số liên kết.

Đối với mỗi hàm, việc xấp xỉ được lưu giữ trong các trọng số và tâm của RBF. Tuy nhiên các trọng số này không phải là duy nhất. RBF có biểu diễn toán học như sau:

$$F(x) = C_0 + \sum_{i=0}^{N-1} C_i \varphi(\|x - R_i\|). \tag{3.1.1}$$

trong đó C - véctơ chứa trọng số RBF. R - véctơ chứa các tâm RBF. φ - hàm cơ sở hoặc hàm kích hoạt của mạng. $F(x)$ - hàm nhận được từ đầu ra của mạng. C_0 - hệ số chệch (có thể là 0), $\|\cdot\|$ - chuẩn euclidean, tức là nếu x chứa n phần tử thì:

$$\| x \| = \sqrt{\sum_{i=1}^{n} x_i^2}. \tag{3.1.2}$$

Mỗi tâm R_j có cùng số chiều với véctơ đầu vào x. Các tâm cũng là các điểm bên trong không gian dữ liệu đầu vào và được chọn sao cho chúng là thể hiện của dữ liệu đầu vào. Khi RBF tính toán quá trình xấp xỉ đối với một số điểm dữ liệu đầu vào thì khoảng cách giữa các điểm đầu vào và mỗi tâm được tính theo khoảng cách euclidean. Những khoảng cách này được chuyển qua φ sau đó được trọng số hóa bằng C_i và được tổng hợp lại để sinh ra đầu ra toàn bộ RBF. Một trong những lựa chọn thông thường nhất đối với hàm cơ sở là hàm Gauss:

$$\varphi(x) = \exp\left(\frac{-x^2}{2\sigma}\right). \tag{3.1.3}$$

trong đó σ là thông số tỷ lệ.

Hàm Gauss thể hiện hình ảnh mỗi đầu ra của hàm cơ sở sẽ xa hơn hoặc gần hơn so với các điểm dữ liệu $x = R_i$ - tâm hàm cơ sở. Mặt khác dạng Gauss của RBF cung cấp các ghép nối qua logic mờ [19] và mỗi vị trí tâm hàm cơ sở có ý nghĩa vật lý nhất định. Xa hơn nữa, mỗi tâm có thể được xem như một dạng hành vi hoặc phản ứng không chỉ của RBF mà còn là của hệ thống mà RBF thực hiện nhận dạng. Ngoài ra có thể có các lựa chọn khác đối với hàm cơ sở như hàm spline sau đây.

$$\varphi(x) = x^2 \ln x. \tag{3.1.4}$$

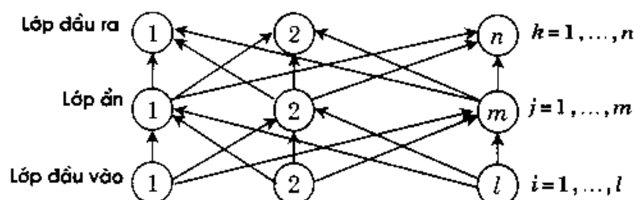
Hàm (3.1.4) làm việc rất hiệu quả trong bài toán nhận dạng mô hình.

3.1.1 Mạng RBF

Hình 13 biểu diễn mạng RBF nhiều đầu vào - nhiều đầu ra. Lớp đầu vào phân bố mỗi thành phần của véctơ đầu vào cho tất cả các nút ẩn. Mỗi nút ẩn trong lớp ẩn chứa một trong

những tâm của RBF và áp hàm cơ sở φ cho khoảng cách euclidean giữa véctơ đầu vào và tâm. Do đó mỗi nút trong lớp ẩn đưa ra một giá trị vô hướng dựa trên tâm mà nút đó có.

Các đầu ra của lớp ẩn được truyền đến lớp đầu ra với các liên kết trọng số. Nút ở lớp đầu ra cộng các đầu vào của nó để tạo ra đầu ra của mạng.



Hình 13 : Mạng RBF

Khi sử dụng mạng RBF trong bài toán nhận dạng, cần lưu ý một vài điểm như sau:

– Dạng φ nào cần chọn ?

– Bao nhiêu tâm sẽ cho kết quả tốt nhất ?

– Tâm cần đặt ở đâu ?

– Bao nhiêu dữ liệu cần thiết đủ để luyện mạng ?

Hàm cơ sở φ có thể có nhiều dạng. Mỗi dạng có thể phù hợp với bài toán này nhưng không phù hợp với bài toán kia. Việc chọn số lượng và vị trí tâm cũng tương tự bài toán chọn số lượng và giá trị ban đầu của trọng số liên kết trong mạng perceptron nhiều lớp. Mặc dù trọng số liên kết của perceptron nhiều lớp có thể được xác định sử dụng thuật lan truyền ngược. Nhưng chưa có thuật nào chọn trọng số ban đầu mà thường chỉ là cho trước ngẫu nhiên. Muốn tìm mô hình tốt nhất cho ĐTĐK thì cần thiết phải tìm số lượng tâm tối ưu. Có quá nhiều tâm hoặc quá ít tâm sẽ cho kết quả không tốt. Nhiều tâm quá sẽ không đủ dữ liệu luyện mạng, nhưng tâm ít quá sẽ cho mô hình sai lệch.

Thuật bình phương nhỏ nhất trong trường hợp có nhiều tâm sẽ tạo ra trọng số liên kết C_i lớn.

Tóm lại vấn đề quan trọng trong khi nhận dạng mô hình bằng mạng RBF là tìm sự cân bằng giữa số lượng tâm và số lượng dữ liệu luyện, giữa số lượng tâm và độ phức tạp của ĐTĐK cần nhận dạng.

Có một số trường hợp đối với tâm của RBF như

– Tâm cố định

– Tâm thay đổi theo cách tự tổ chức

– Tâm được chọn từ thuận học có giám sát. Ví dụ thuật giảm gradient.

Nếu tâm cố định thì chúng có thể được chọn từ không gian dữ liệu đầu vào. Có nhiều cách đặt vị trí tâm trong không gian này tùy theo phân bố của dữ liệu. Nếu không có thông tin ban đầu về phân bố, có thể coi dữ liệu có phân bố đều hay phân bố ngẫu nhiên. Phân bố tốt nhất phụ thuộc vào bài toán riêng biệt với dạng ĐTĐK cụ thể và dữ liệu cụ thể.

3.1.2 Luyện mạng RBF

Việc luyện mạng RBF phụ thuộc vào việc chọn tâm như thế nào. Có 2 kỹ thuật luyện mạng RBF.

1. Chọn các giá trị của tâm cố định. Sau đó sử dụng kỹ thuật thích nghi để luyện mạng tìm ra các trọng số C_i tối ưu.

2. Giá trị của tâm không được chọn cố định mà được chọn trong quá trình luyện. Như vậy cả R_i và C_i được tìm trên cơ sở sử dụng các phương trình giảm gradient.

Kỹ thuật 2) nói chung chậm hơn kỹ thuật 1). Nhưng nếu tập dữ liệu bị giới hạn thì kỹ thuật 2) sẽ cho kết quả tốt hơn kỹ thuật 1).

3.1.3 Mở rộng mạng RBF - Mạng RBF nhiều lớp

Mạng RBF và mạng perceptron nhiều lớp có một số ưu và nhược điểm như sau: Mạng RBF có tốc độ hội tụ nhanh nhưng có thể bị mất thông tin do khả năng tổng hợp thấp nếu sử dụng thuật lan truyền ngược sai số.

Mạng perceptron nhiều lớp thường có tốc độ hội tụ chậm nhưng độ chính xác cao trong nhận dạng. Như vậy cần mở rộng mạng RBF theo hướng thêm các lớp ẩn bên trong RBF để hội tụ được những đặc trưng tốt nhất của cả hai kiểu mạng. Có nghĩa là nâng cao độ chính xác trong quá trình nhận dạng mô hình nhưng vẫn đảm bảo hội tụ nhanh.

Ngoài ra khi mạng RBF được luyện sử dụng thuật bình phương nhỏ nhất thì RBF hoạt động nhanh hơn perceptron nhiều lớp. Nhưng nếu tập mẫu để luyện được chọn kém thì mạng RBF lại không cho kết quả tối ưu toàn cục tốt như mạng perceptron nhiều lớp. Một số kết quả đã đạt được trong việc mở rộng mạng RBF như [10]. Sự khác nhau giữa RBF nhiều lớp so với perceptron nhiều lớp là ở các phương trình nút tại các lớp ẩn và lớp đầu ra. Trong mạng RBF nhiều lớp không sử dụng thuật luyện tối ưu theo bình phương nhỏ nhất.

3.2 Nhận dạng mô hình

Nhận dạng mô hình là quá trình xác định mô hình của ĐTĐK về cấu trúc và thông số trên cơ sở đầu vào và đầu ra của ĐTĐK. Thông thường mô hình hồi quy được sử dụng. Ở đây động học ĐTĐK được xét dưới dạng hệ rời rạc với véctơ đầu vào bao gồm:

$$x=(y_{t-1}, y_{t-2}, \ldots, u_{t-1}, u_{t-2}, \ldots), \qquad (3.2.1)$$

trong đó y_{t-1} và u_{t-1} là các giá trị đầu ra và đầu vào tương ứng ở chu kỳ trước.

Lưu ý rằng véctơ đầu vào mạng nơron sẽ hoàn toàn bao gồm các giá trị đầu ra và đầu vào ĐTĐK. Tuy nhiên có thể tồn tại một vài khả năng khác. Trong [13] sử dụng véctơ đầu vào mạng dưới dạng:

$$x = (v_{t-1}, v_{t-2}, \ldots, u_{t-1}, u_{t-2}, \ldots), \tag{3.2.2}$$

ở đây v_{t-1} là giá trị đầu ra của mạng RBF, có nghĩa là $F(x)$ ở chu kỳ trước.

Mô hình thu được sau khi nhận dạng gọi là tốt nếu nó thể hiện được đúng ĐTĐK. Như vậy có thể sử dụng mô hình thay cho ĐTĐK để dự báo, kiểm tra và điều khiển. Mạng nơron được luyện để mô hình hóa quan hệ vào - ra của ĐTĐK. Như vậy quy trình nhận dạng mô hình có bản chất là thuật toán luyện mạng.

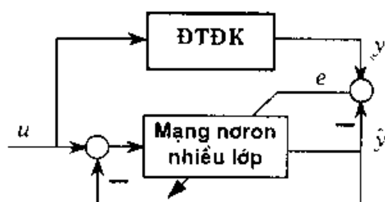
Mạng RBF cho nhận dạng mô hình được xem xét kỹ ở [3,9].

Biến thể của mạng RBF là mạng Gauss xuyên tâm (radial gaussian network) được sử dụng để nhận dạng hệ phi tuyến [32].

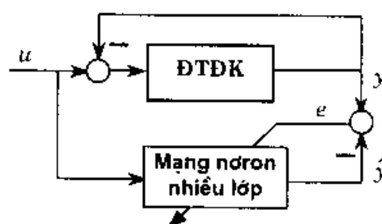
Mạng perceptron nhiều lớp dùng cho nhận dạng mô hình với nhiều thuật toán khác nhau được xem xét trong [6].

Mạng spline cũng được sử dụng nhiều cho nhận dạng mô hình với yêu cầu về bộ nhớ máy tính đủ lớn. Nhìn chung mạng nơron nhiều lớp sử dụng cho nhận dạng mô hình phi tuyến theo hai quan điểm:

1) Sử dụng mạng nhiều lớp như mô hình song song với ĐTĐK. Mạng được luyện sử dụng sai số giữa đầu ra mô hình và đầu ra của ĐTĐK. Phản hồi lấy từ đầu ra của mạng như ở hình 14.
2) Sử dụng mạng nhiều lớp như mô hình liên tiếp - song song. Mạng cũng được luyện sử dụng sai số giữa đầu ra của mạng và đầu ra của ĐTĐK. Nhưng phản hồi lấy từ đầu ra của ĐTĐK như hình 15.



Hình 14 : Mô hình song song.



Hình 15 : Mô hình liên tiếp - song song.

Mô hình liên tiếp - song song có ưu thế hơn so với mô hình song song vì thuật lan truyền ngược sai số có thể sử dụng bình thường do không có vòng phản hồi trong mạng nơron. Nếu ĐTĐK được giả thiết là ổn định với đầu vào đầu ra bị giới hạn (Bounded input Bounded output stable) thì mọi đầu vào mạng nơron bị giới hạn và khi sai số đầu ra tiệm cận đến giá trị đủ nhỏ, mô hình song song có thể thay thế bằng mô hình liên tiếp - song song mà không có hệ quả nào nghiêm trọng xảy ra.

Cấu trúc mạng nơron giải bài toán nhận dạng mô hình rất đa dạng [8], tùy thuộc vào từng bài toán cụ thể.

Mạng RBF khi sử dụng trong nhận dạng mô hình cần lưu ý một số ưu điểm và nhược điểm sau đây:

Ưu: Quá trình học tương đối đơn giản và nhanh chóng so với mạng perceptron nhiều lớp.

Nhược: Khi véctơ đầu vào có số chiều lớn sẽ kéo theo số tâm lớn và như vậy khối lượng tính toán nhiều và có thể sinh ra hiệu ứng không chỉnh làm sai lệch quá trình nhận dạng mô hình. Ngoài ra, nếu tính ngẫu nhiên của dữ liệu tăng sẽ dẫn đến việc làm tăng số lượng tâm và như vậy phải tăng thêm chiều cho véctơ dữ liệu đầu vào để đảm bảo độ chính xác nhận dạng mô hình. Khối lượng tính toán tăng lên làm cho thời gian nhận dạng bị kéo dài.

3.3 Ví dụ nhận dạng hệ động học phi tuyến sử dụng mạng RBF (26)

Xét hệ động học phi tuyến của ĐTĐK

$$\dot{x} = f(x) + g(x)u \qquad (3.3.1)$$

Giả thiết ĐTĐK là ổn định theo vòng hở. Véctơ trạng thái x quan sát được. Số chiều n của x biết trước. Cần tìm mô hình xấp xỉ ĐTĐK (3.3.1).

Rõ ràng có thể viết (3.3.1) dưới dạng sau:

$$\dot{x} = Ax + (f(x) - Ax) + g(x)u \qquad (3.3.2)$$

Ở đây $A \in R^{n \times m}$ là ma trận ổn định cho trước.

Theo tính chất xấp xỉ của mạng RBF cho các hàm phi tuyến: Nếu số lượng các nút trong lớp ẩn là đủ lớn, thì $f(x) - Ax$ và $g(x)$ có thể xấp xỉ bằng các mạng RBF sau:

$$f(x) - Ax = W^* S(x) \quad \text{và} \quad g(x) = V^* S(x),$$

trong đó $W^* \in R^{n \times N}$ và $V^* \in R^{n \times N}$ là các ma trận trọng số của các tổ hợp tuyến tính trên. N xác định số lượng nút trong một lớp RBF của mạng.

$$S(x) = [S_1, S_1, \ldots, S_N]^T,$$

véctơ các hàm cơ sở sau:

$$S_i = \left(\|x - C_i\|^2 + \rho_i^2 \right)^{-\frac{1}{2}}, \quad \forall\ i,\ i = 1, 2, \ldots, N.$$

Tâm $C_i \in \mathbb{R}^n$ và độ rộng $\rho_i \in \mathbb{R}^n$ được biết trước. Như vậy phương trình (3.3.1) có thể viết dưới dạng sau:

$$\dot{x} = Ax + W^* S(x) + V^* S(x) u. \tag{3.3.3}$$

Dựa trên (3.3.3) mô hình ước lượng của ĐTĐK có thể được mô tả bằng phương trình sau:

$$\dot{\tilde{x}} = A\tilde{x} + WS(x) + VS(x)u. \tag{3.3.4}$$

ở đây $W \in \mathbb{R}^{n \times N}$, $V \in \mathbb{R}^{n \times N}$ là các ma trận ước lượng của W^* và V^* tương ứng, $\tilde{x} \in \mathbb{R}^n$ là ước lượng trạng thái của x. Gọi

$$x_e = \tilde{x} - x, \quad W_e = W - W^*, \ V_e = V - V^*,$$

phương trình sai số ước lượng sẽ là:

$$\dot{x}_e = Ax_e + W_e S(x) + V_e S(x) u. \tag{3.3.5}$$

Thuật nhận dạng sử dụng hàm Lyapunov sau đây để đảm bảo tính hội tụ:

$$L(x_e, W_e, V_e) = \frac{1}{2} x_e^T P x_e + \frac{1}{2} Tr(W_e^T W_e) + \frac{1}{2} Tr(V_e^T V_e), \tag{3.3.6}$$

ở đây P là ma trận đối xứng xác định dương. Có thể xác định ma trận Q đối xứng xác định dương thỏa mãn phương trình Lyapunov sau:

$$PA + A^T P = -Q.$$

Thay (3.3.5) vào (3.3.6) và xét đạo hàm của hàm Lyapunov, thu được:

$$\dot{L} = \frac{1}{2} x_e^T (PA + A^T P) x_e + S^T(x) W_e^T P x_e + S^T(x) V_e^T P x_e u +$$
$$+ Tr(\dot{W}_e^T W_e) + \frac{1}{2} Tr(\dot{V}_e^T V_e). \tag{3.3.7}$$

Nếu chọn

$$Tr(\dot{W}_e^T W_e) = -S^T(x) W_e^T P x_e, \tag{3.3.8}$$

$$Tr(\dot{V}_e^T V_e) = -S^T(x) V_e^T P x_e u, \tag{3.3.9}$$

thì:

$$\dot{L}(x_e, W_e, V_e) = -\frac{1}{2} x_e^T Q x_e. \tag{3.3.10}$$

Do ma trận sai số W^* và V^* là ma trận hằng, nên từ (3.3.8), (3.3.9) nhận được thuật nhận dạng mô hình sau đây:

$$\dot{W}_{ij} = -S_j \sum_{l=1}^{n} P_{il} x_{el} \qquad (3.3.11a)$$

$$\dot{V}_{ij} = -S_j \sum_{l=1}^{n} P_{il} x_{el} u \qquad (3.3.11b)$$

với $i = 1, 2, \ldots, N$ và $j = 1, 2, \ldots, N$, P_{ij} là phần tử của ma trận Lyapunov P.

Từ (3.3.6) thấy rằng $L(x_e, W_e, V_e) \geq 0$.

Từ (3.3.10) nhận được $\dot{L}(x_e, W_e, V_e) \leq 0$.

Vì vậy $x_e(t) \to 0$, $W_e \to 0$, $V_e \to 0$, hoặc $\hat{x} \to x$, $W \to W^*$, $V \to V^*$, khi $t \to \infty$.

Để tính toán đơn giản có thể chọn

$A = aI, Q = qI$ và $P = pI$ với $a > 0$, $q > 0$ và I là ma trận đơn vị.

Khi đó thuật nhận dạng mô hình đơn giản như sau:

$$\dot{W}_{ij} = -pS_j x_{ei} \, ,$$

$$\dot{V}_{ij} = -pS_j x_{ei} u \, .$$

Từ phương trình Lyapunov rút ra:

$$p = \frac{q}{2a} > 0 \, .$$

Để hội tụ đến trọng số thực, hệ động lực phải có đủ giàu thông tin ở đầu vào. Vì thế đa số đầu vào phải được chọn ngẫu nhiên.

3.4 Ví dụ về điều khiển thích nghi sử dụng mạng RBF (7)

Mạng nơron nhiều lớp có khả năng xấp xỉ mọi hàm phi tuyến và được sử dụng rộng rãi trong nhận dạng và điều khiển [15, 23, 43]. Mạng nơron có thể xem như một hệ phi tuyến với đầu ra là biến đổi phi tuyến của đầu vào và trọng số liên kết. Như vậy bài toán nhận dạng thông số mô hình trong điều khiển thích nghi phải dựa trên cơ sở kỹ thuật tối ưu phi tuyến. Kỹ thuật này đòi hỏi tính toán phức tạp mà thường chỉ đạt đến tối ưu địa phương. Trong những ứng dụng cụ thể thường hàm kích hoạt sigmoid được chọn và thuật toán lan truyền ngược là phương pháp luyện mạng cơ bản. Nhưng phương pháp này có tốc độ hội tụ chậm và cũng không đảm bảo tối ưu toàn cục.

Mạng RBF có ưu điểm là đầu ra có quan hệ tuyến tính đối với trọng số liên kết. Phương pháp luyện nhanh và hiệu quả. Những ưu điểm này được nhấn mạnh trong [22].

Điều khiển sử dụng mạng nơron RBF được đề cập trong nhiều công trình [11.27].

Trong mọi phương pháp điều khiển thích nghi thì chiến lược điều khiển thích nghi dựa trên mô hình nghịch là chiến lược điều khiển khá đơn giản nhưng hiệu quả. Nếu đưa mạng RBF vào cấu trúc điều khiển sẽ nâng cao hiệu quả hơn nữa. Hàm kích hoạt có thể chọn các hàm spline cho bài toán nhận dạng động học nghịch của ĐTĐK. Với những ý tưởng này, xét mạng RBF với n đầu vào và m đầu ra dưới dạng véctơ:

$$f_r(x)=c_0+\sum_{j=1}^{M} C_j\varphi(\|x-R_j\|) \tag{3.4.1}$$

ở đây $x\in R^n$, $C_j\in R^m$, $C_0\in R^m$, $R_j\in R^n$, $j=1,2, \ldots ,M$, $\varphi(.)$ là hàm và $\|\cdot\|$ là chuẩn euclidean. Giả sử:

$$C=[C_{mj}]^T,\ j=1,2,\ \ldots\ ,M. \tag{3.4.2}$$

Biểu thức (3.4.1) dưới dạng vô hướng được viết như sau:

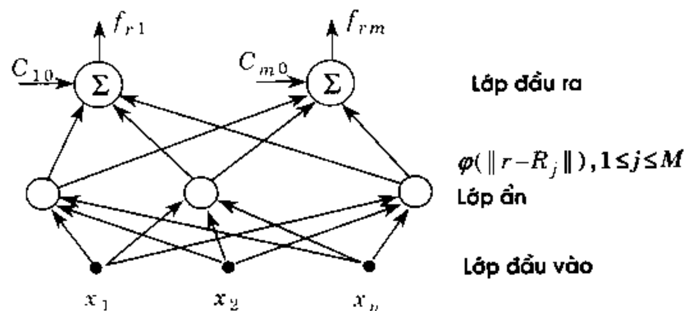
$$f_{ri}(x)=c_{i0}+\sum_{j=1}^{M} C_{ij}\varphi(\|x-R_j\|),\ i=1,2,\ \ldots\ ,m. \tag{3.4.3}$$

Nếu $\varphi(.)$ và các tâm R_j là cố định, tập đầu vào $x(t)$ và đầu ra mong muốn $d(t)$, $t=1,2, \ldots ,N$, được cho trước thì các trọng số liên kết C_{ij} với $i=1,2, \ldots ,m$, và $j=1,2, \ldots ,M$ sẽ được xác định trên cơ sở thuật bình phương nhỏ nhất tuyến tính và ở đây không tồn tại bài toán cực tiểu địa phương vì mô hình RBF là mạng nơron có đầu ra tuyến tính theo trọng số liên kết. Biểu diễn mạng RBF trên hình 16.

Hàm $\varphi(.)$ có thể chọn như sau:

$$\varphi(z)=z^2\ln z. \tag{3.4.4}$$

Cần nhấn mạnh rằng quá trình xử lý thông tin của RBF phụ thuộc vào các tâm R_j cho trước. Các tâm này phải được cho phù hợp với miền đầu vào $\{x(t)\}^N$. Lưu ý đến một số cách chọn tâm, ví dụ [5].



Hình 16 : Mạng RBF

Giả sử hệ động học phi tuyến ổn định theo giới hạn đầu vào - giới hạn đầu ra (BIBO Stable) và được mô tả bằng biểu thức sau:

$$y(k+1)=f[y(k),y(k-1),\dots,y(k-n+1),u(k-\theta),u(k-\theta-1),\dots,u(k-\theta-m)], \quad (3.4.5)$$

trong đó y là đầu ra của ĐTĐK, u là đầu ra của Bộ điều khiển, n và m là các trễ cực đại ở đầu ra và đầu vào tương ứng, θ là trễ thời gian biết trước.

Nếu hệ thống có động học nghịch (invertible) thì mô hình động học nghịch có thể biểu diễn dưới dạng sau:

$$u(k-\theta)=f[y_e(k+1),y(k),\dots,y(k-n+1),\ u(k-\theta-1),\dots,u(k-\theta-m)], \quad (3.4.6)$$

ở đây $y_e(k+1)$ là đầu ra kỳ vọng tại bước $(x+1)$ của ĐTĐK.

Gọi

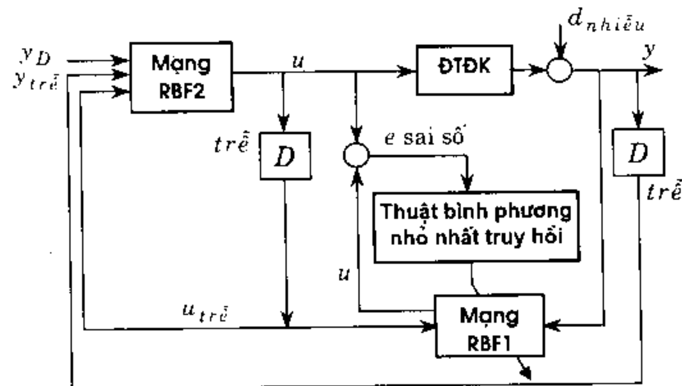
$$x(k)=[y_e(k+1),y(k),\dots,y(k-n+1),u(k-\theta-1),\dots,u(k-\theta-m)], \quad (3.4.7)$$

như vậy có thể sử dụng mạng RBF để mô hình hóa động học nghịch như sau:

$$\hat{u}(k-\theta)=C_0+\sum_{i=1}^{\overline{M}}C_i\varphi(\|x(k)-R_i\|),\ x(k)\in R^{m-n}, M<N, \quad (3.4.8)$$

trong đó $\hat{u}(k-\theta)$ là ước lượng đầu vào tại bước k tương ứng với đầu ra kỳ vọng $y_e(k+1)$ và N là số lượng các mẫu đầu vào.

Bộ điều khiển thích nghi dựa trên mạng RBF được thiết kế cụ thể như hình 17.



Hình 17: Sơ đồ điều khiển dựa trên 2 mạng RBF.

Có 2 mạng RBF1 và RBF2 trong sơ đồ điều khiển. RBF2 mô hình hóa động học nghịch của ĐTĐK và thường được sử dụng như bộ điều khiển. Còn RBF1 có cấu trúc giống hệt RBF2. Khi RBF1 tại thời điểm bắt đầu của điều khiển, được dùng như đơn vị phản hồi.

Trong sơ đồ điều khiển trên sử dụng thuật bình phương nhỏ nhất truy hồi để hiệu chỉnh thông số mạng.

y_D là đầu ra kỳ vọng của ĐTĐK ở bước tiếp theo thời điểm hiện tại được đưa vào RBF2. Ngoài ra còn 2 đầu vào khác được lấy từ đầu ra RBF2 qua bộ trễ và từ đầu ra của ĐTĐK cũng qua bộ trễ. Đầu ra RBF2 là tín hiệu điều khiển u. Mục tiêu điều khiển là tạo ra đầu ra kỳ vọng y_D. Tuy nhiên rất khó đạt được mục tiêu khi có tác động bất thường của nhiễu môi trường d hoặc bất định trong ĐTĐK. Chính vì vậy mạng RBF1 được dùng đến để giải quyết khó khăn này. Sự sai lệch e giữa đầu ra RBF2 (đầu vào ĐTĐK) và đầu ra của RBF1 sẽ được sử dụng để hiệu chỉnh trọng số của mạng RBF1 theo thuật bình phương nhỏ nhất truy hồi. Các trọng số mới của RBF1 được sao chép sang mạng RBF2 đảm bảo cấu trúc giống nhau giữa RBF1 và RBF2. Thuật toán dừng khi sai số e đạt giá trị cực tiểu. Khi đó đầu vào và đầu ra của RBF1 chính là đầu vào và đầu ra của RBF2 $(y = y_{\text{trễ}})$. Như vậy toàn bộ quá trình điều khiển gồm hai bước sau đây:

a) Nhận dạng off-line

b) Điều khiển thích nghi on-line

Trong quá trình nhận dạng mô hình, các tâm RBF và các trọng số ban đầu cho trước. Trong khi điều khiển on-line, các tâm RBF không thay đổi, các trọng số được hiệu chỉnh theo sự thay đổi của môi trường và độ bất định của hệ thống (ĐTĐK). Để thu được điều khiển tốt hơn có thể cải tiến thuật luyện mạng như sau: Coi biểu thức (3.4.8) là trường hợp đặc biệt của mô hình hồi quy tuyến tính:

$$d(t) = \sum_{i=1}^{M} p_i(t)\theta_i + e(t), \tag{3.4.9}$$

ở đây $d(t)$ - đầu ra mong muốn: θ - các trọng số

$P_i(t)$ - bộ hồi quy (regressors) và là một hàm nào đó cố định của $x(t)$

$$P_i(t) = P_i(x(t)) = \varphi\|x(t) - R_j\|) \tag{3.4.10}$$

sai số $e(t)$ giả thiết không tương quan với $P_i(t)$. Như vậy bài toán chỉnh trọng số có thể hiểu trên cơ sở ý tưởng cải biên thuật toán bình phương nhỏ nhất truy hồi sau đây:

$$\hat{c}(k+1) = \hat{c}(k) + \Delta c(k) \tag{3.4.11}$$

Từ đây có thể xây dựng một kiểu thuật toán cải biên như sau:

$$\hat{c}(k+1) - \hat{c}(k) = k_p \Delta c(k) + k_d(\Delta c(k) - \Delta c(k-1)) + \frac{1}{k_i}\sum_{j=1}^{k} \Delta c(j). \tag{3.4.12}$$

Quy tắc (3.4.12) tương đương với quy tắc điều khiển PID trong lý thuyết điều khiển. Đại lượng đầu của vế phải (3.4.12) là đại diện tỷ lệ thuận với đại lượng cải biên $\Delta c(k)$ thu

được từ thuận toán bình phương nhỏ nhất truy hồi áp dụng cho (3.4.9). Đại lượng thứ hai tỷ lệ với tốc độ thay đổi của đại lượng cải biên đó. Đại lượng thứ ba tỷ lệ với tổng các đại lượng cải biên đã có trước đây. So sánh (3.4.11) và (3.4.12), có thể thấy rằng (3.4.11) có thể thu được từ (3.4.12) nếu

$$k_p = 1, \qquad k_d = 0, \qquad k_i = \infty.$$

Thuật toán này gọi là thuật bình phương nhỏ nhất truy hồi PID [7]. Thuật toán này hội tụ nhanh hơn thuật toán bình phương nhỏ nhất thông thường.

3.5 Mạng nơron nhiều lớp và một số thuật học trong nhận dạng mô hình và điều khiển

Mạng nơron nhiều lớp được sử dụng khá rộng rãi cho các bài toán nhận dạng ảnh, nhận dạng mô hình và điều khiển. Thuật học lan truyền ngược sai số là một thuật học tương đối phổ biến cho các bài toán trên trong những năm 90. Thuật học này đã được Werbos P.J. xây dựng từ 1974 [42]. Tuy nhiên nhiều nhà khoa học thực tế cho rằng tiệm cận trên thuộc về các công trình của nhóm tác giả về tính toán song song [30, 31].

Quá trình học của mạng nơron nhiều lớp được thực hiện bằng cách so sánh các đầu ra của mạng với các tín hiệu chỉ đạo, thiếu vắng thông tin về các đầu ra ở các lớp ẩn. Sự hiểu biết về cấu trúc mạng nhiều lớp cho phép tính toán các tín hiệu hiệu chỉnh trọng số liên kết không chỉ đối với những nơron lớp cuối cùng mà còn đối với cả các lớp ẩn. Một số tác giả cải tiến phương pháp lan truyền ngược sai số để nâng cao độ chính xác về tốc độ tính toán [12,14]. Một số phương pháp khác làm tăng tốc độ hội tụ của quá trình học thông qua việc sử dụng các thuật toán dạng Newton, trong đó ngoài việc thể hiện gradient của phiếm hàm cần cực tiểu còn cần phải có cả ma trận đạo hàm bậc hai. Vì vậy các thuật toán này gọi là thuật toán học bậc hai [4]. Các thuật học dạng lan truyền ngược sai số thuộc dạng thuật toán Newton và có thể thể hiện trong mạng nơron với cỡ không lớn lắm. Đối với mạng cỡ lớn thuật lan truyền ngược đòi hỏi khối lượng tính toán lớn để xác định gradient bậc nhất và đối với một số dạng thuật học nâng cao đòi hỏi xác định gradient bậc hai.

3.5.1 Cấu trúc mạng nơron nhiều lớp

Hình 18 mô tả mạng nơron phổ biến nhiều lớp thực hiện chức năng biến đổi phi tuyến với liên kết trọng số, trong đó:

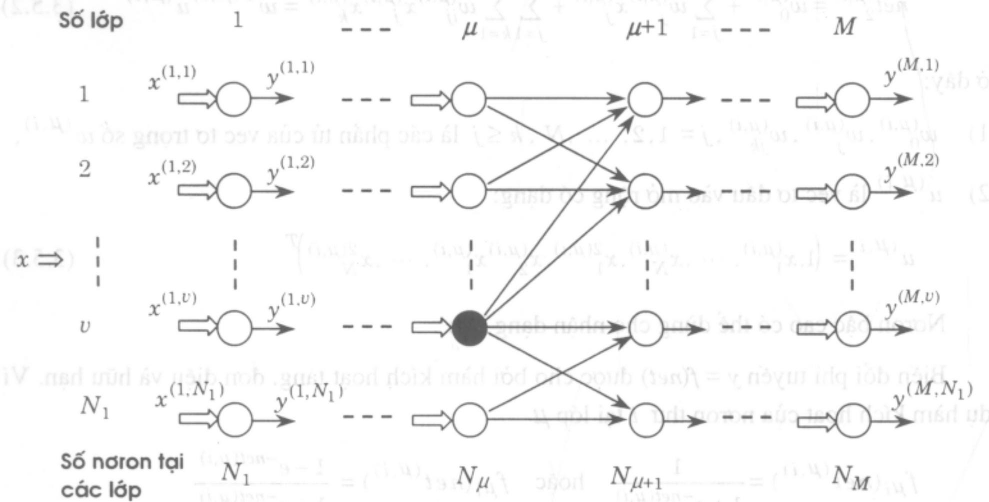
1) M – số lớp của mạng.
2) N_μ – số nơron tại lớp μ và không có mối liên kết giữa các nơron cùng một lớp.
3) Để bài toán trở nên đơn giản nhưng không mất tính tổng quát cần quan niệm như sau: Những đầu ra của nơron lớp μ, $\mu = 1, 2, \ldots, M-1$ chỉ liên kết với lớp tiếp theo $(\mu+1)$.

4) Cấu trúc liên kết giữa các nơron lớp μ và (μ+1) thể hiện bằng ma trận $C^{(\mu)}$ với 2 giá trị 0 và 1:

a) Nếu $C_{i,j}^{(\mu)} = 1$ thì đầu ra của nơron thứ i tại lớp μ được đưa vào nơron thứ j tại lớp (μ+1).

b) Nếu $C_{i,j}^{(\mu)} = 0$ thì không có liên kết đó.

5) x – vectơ tín hiệu vào từ bên ngoài chỉ tác động lên đầu vào các nơron lớp đầu tiên (μ=1). Các tác động của tín hiệu bên ngoài vào lớp nơron đầu tiên được biểu diễn bằng ma trận tác động $C^{(0)}$ có cấu trúc tương tự $C^{(\mu)}$. Các đầu ra của lớp nơron cuối cùng (μ=M) tạo thành vectơ đầu ra $y^{(M)}$.



Hình 18 : Cấu trúc mạng nơron nhiều lớp

Mỗi nơron thứ i tại lớp μ biến đổi vectơ đầu vào $x^{(\mu,i)}$ thành đại lượng vô hướng $y^{(\mu,i)}$. Biến đổi này gồm 2 giai đoạn: đầu tiên tính toán hàm $net^{(\mu,i)}$, sau đó biến đổi hàm này thành đầu ra vô hướng $y^{(\mu,i)} = f_{\mu i}(net^{(\mu,i)})$ với hàm kích hoạt $f_{\mu i}$ của nơron i tại lớp μ. Hàm $net^{(\mu,i)}$ là một đoạn chuỗi Taylor nhiều chiều; bậc cao nhất của hàm này là bậc của nơron. Nếu bậc bằng 1, tương ứng với nơron bậc nhất; bậc từ 2 trở đi tương ứng với nơron bậc cao. Các hệ số phân rã chuỗi Taylor nhiều chiều tạo thành vectơ trọng số liên kết $w^{(\mu,i)}$ hay còn gọi là bộ nhớ của nơron. Đối với bài toán nhận dạng mô hình và điều khiển, ít khi sử dụng nơron bậc cao, chủ yếu là hàm bậc 1:

$$net_1^{(\mu,i)} = w_0^{(\mu,i)} + \sum_{j=1}^{N} w_j^{(\mu,i)} x_j^{(\mu,i)} = w^{T(\mu,i)} u^{(\mu,i)} \tag{3.5.1}$$

trong đó:

1) $w^{(\mu,i)} = \left(w_0^{(\mu,i)}, w_1^{(\mu,i)}, \cdots, w_N^{(\mu,i)}\right)^T$ là vec tơ trọng số liên kết với nơron,

2) $u^{(\mu,i)} = \left(1, x_1^{(\mu,i)}, \cdots, x_N^{(\mu,i)}\right)^T$ là vec tơ mở rộng đầu vào nơron thứ i tại lớp μ (ngưỡng được coi là đầu vào 1).

3) $x_j^{(\mu,i)}$ là phần tử j của vec tơ đầu vào $x^{(\mu,i)}$ có chiều N.

Đối với nơron bậc hai:

$$net_2^{(\mu,i)} = w_0^{(\mu,i)} + \sum_{j=1}^{N} w_j^{(\mu,i)} x_j^{(\mu,i)} + \sum_{j=1}^{N} \sum_{k=1}^{j} w_{ij}^{(\mu,i)} x_j^{(\mu,i)} x_k^{(\mu,i)} = w^{T(\mu,i)} u^{(\mu,i)} \tag{3.5.2}$$

ở đây:

1) $w_0^{(\mu,i)}, w_j^{(\mu,i)}, w_{jk}^{(\mu,i)}, j = 1, 2, \ldots, N, k \le j$ là các phần tử của vec tơ trọng số $w^{(\mu,i)}$.

2) $u^{(\mu,i)}$ là vec tơ đầu vào mở rộng có dạng:

$$u^{(\mu,i)} = \left(1, x_1^{(\mu,i)}, \cdots, x_N^{(\mu,i)}, x_1^{2(\mu,i)}, x_2^{(\mu,i)} x_1^{(\mu,i)}, \cdots, x_N^{2(\mu,i)}\right)^T \tag{3.5.3}$$

Nơron bậc cao có thể dùng cho nhận dạng ảnh.

Biến đổi phi tuyến $y = f(net)$ được cho bởi hàm kích hoạt tăng, đơn điệu và hữu hạn. Ví dụ hàm kích hoạt của nơron thứ i tại lớp μ

$$f_{\mu_i}(net^{(\mu,i)}) = \frac{1}{1 + e^{-net(\mu,i)}} \quad \text{hoặc} \quad f_{\mu_i}(net^{(\mu,i)}) = \frac{1 - e^{-net(\mu,i)}}{1 + e^{-net(\mu,i)}}$$

3.5.2 Mối liên hệ giữa đầu vào và đầu ra của mạng nơron nhiều lớp bậc nhất

$net_1^{(\mu,i)} = net^{(\mu,i)}$ được biểu diễn qua (3.5.1)

Ở mạng nơron nhiều lớp với đầy đủ liên kết, đầu ra của nơron i tại lớp μ được mô tả bằng phương trình sau:

$$y^{(\mu,i)} = f\left(w_0^{(\mu,i)} + \sum_{j=1}^{N} w_j^{(\mu,i)} y_j^{(\mu-1,j)}\right) = f(w^{T(\mu,i)} u^{(\mu,i)})$$
$$= f(net^{(\mu,i)}), \qquad j = 1, 2, \ldots, N_\mu \tag{3.5.4}$$

trong đó $u^{(\mu,i)}$ là vec tơ đầu vào mở rộng (kể cả ngưỡng) của nơron i tại lớp μ

Để viết gọn, sử dụng các ký hiệu sau đây:

a) $y^{(\mu)} = (y^{(\mu,1)}, y^{(\mu,2)}, \dots, y^{(\mu,N_\mu)})^T$ (3.5.5)

là vec tơ đầu ra của các nơron tại lớp μ có số nơron N_μ

b) $W^{(\mu)} = (w^{(\mu,1)}, w^{(\mu,2)}, \dots, w^{(\mu,N_\mu)}) = \left(w_0^{(\mu)}, W_1^{(\mu)}\right)$ (3.5.6)

là ma trận trọng số liên kết có chiều N_μ ($N_{\mu+1}$+1) của các nơron tại lớp μ

c) $w_0^{(\mu)} = \left(w_0^{(\mu,1)}, w_0^{(\mu,2)}, \cdots, w_0^{(\mu,N_\mu)}\right)^T$

là vec tơ có chiều N_μ bao gồm các trọng số liên kết ngưỡng $w_0^{(\mu,j)}$

Các ký hiệu a), b) và c) cho phép viết (3.5.4) dưới dạng sau đây:

$$y^{(\mu,t)} = f\left(w_0^{(\mu)} + W_1^{(\mu)} y^{(\mu-1)}\right) = f^{(\mu)}(y^{(\mu-1)}), \mu = 1, 2, \dots, M \quad (3.5.8)$$

trong đó $f^{(\mu)}$ là vec tơ hàm phi tuyến biến đổi từng vec tơ thành phần $w_0^{(\mu)} + W_1^{(\mu)} y^{(\mu-1)}$ tương ứng với quan hệ (3.5.4).

Phương trình hồi quy phi tuyến (3.5.8) cho phép lập mối quan hệ giữa đầu vào và đầu ra của mạng nơron nhiều lớp dưới dạng tổng quát sau:

$$Y^{(M)} = f^{(M)}\left(w_0^{(M)} + W_1^{(M)} f^{(M-1)}\left(w_0^{(M-1)} + W_1^{(M-1)} f^{(M-2)}(\cdots\right.\right.$$

$$\cdots\left(w_0^{(2)} + W_1^{(2)} f^{(1)}\left(w_0^{(1)} + W_1^{(1)} x\right) \cdots\right) = f^{(M)} f^{(M-1)} \dots f^{(1)} x. \quad (3.5.9)$$

Như vậy đầu ra của mạng nơron nhiều lớp là vec tơ hàm phi tuyến phức tạp của vec tơ đầu vào x và vec tơ hàm này phụ thuộc vào vec tơ trọng số liên kết trong mạng.

3.5.3 Thuật học của mạng nơron nhiều lớp và phương pháp lan truyền ngược sai số

Quá trình tìm các giá trị tối ưu của trọng số liên kết là quá trình học của mạng và quá trình này được thực hiện nhằm cực tiểu tiêu chuẩn chất lượng nào đó $J(w)$ đặc trưng cho độ đo gần nhau được tích hợp ở các đầu ra của mạng $y^{(M)}(n)$ và tín hiệu chỉ đạo $y^*(n)$. Ví dụ:

$$J(w) = \frac{1}{n} \sum_{m=1}^{n} Q(e(w, m)) \quad (3.5.10)$$

n – là thời điểm ra của mạng nơron và tín hiệu chỉ đạo tương ứng.

w - vec tơ của các vec tơ thành phần biểu diễn trọng số liên kết của mạng

$$w = (w^{(M)T}, w^{(M-1)T}, \dots, w^{(1)T})^T. \quad (3.5.11)$$

Ở đây các vec tơ thành phần $w^{(\mu)}$, $\mu=\overline{M,1}$ chứa các vec tơ $w^{(\mu,i)}$, $i=\overline{1,N_\mu}$ biểu diễn trọng số liên kết với nơron i tại lớp μ:

$$w^{(\mu)}=(w^{(\mu,1)T},w^{(\mu,2)T},\ \dots\ ,w^{(\mu,N_\mu)T})^T. \tag{3.5.12}$$

Tiêu chuẩn chất lượng học tức thời $Q(e(w,n))$ phụ thuộc vào vec tơ sai số mạng $e(w,m)$

$$e(w,n)=y^{(M)}(m)-y^*(m). \tag{3.5.13}$$

Trong các bài toán nhận dạng và điều khiển thường Q có dạng bình phương:

$$Q(e(w,n))=e^T(w,n)Re(w,n), \tag{3.5.14}$$

với R là ma trận xác định dương.

Cực tiểu phiếm hàm tiêu chuẩn tích hợp chất lượng học $J(w)$ được thực hiện chủ yếu nhờ phương pháp gradient. Như vậy sẽ đưa đến biểu thức tìm trọng số dưới đây:

$$w(n)=w(n-1)-\Upsilon\nabla_w Q(e(w(n-1),n)) \tag{3.5.15}$$

trong đó:

$\nabla_w Q(\cdot)=\dfrac{\partial Q(e(w(n-1),n))}{\partial w}$ là gradient của phiếm hàm tức thời $Q(\cdot)$ đối với vec tơ trọng số liên kết của mạng

Đặc trưng Υ hoặc là hằng và khi đó thông thường chuỗi $w(n)$ hội tụ đến lân cận giá trị tối ưu w, hoặc Υ là hàm giảm theo thời gian, ví dụ như thuật toán tối ưu thích nghi ngẫu nhiên [38].

Ở đây sẽ không bàn đến vấn đề hội tụ của (3.5.15), chỉ lưu ý rằng (3.5.15) có tốc độ hội tụ chậm và nhiều khả năng làm phiếm hàm (3.5.10) rơi vào tối ưu địa phương. Từ quan điểm tính toán, khó khăn cơ bản thực hiện thuật học (3.5.15) là việc tìm gradient của hàm phức tạp $Q(e(w,m))$. Chính vì vậy, các tác giả [30, 31] đã tìm được thủ tục có hiệu quả để tính toán gradient của hàm $Q(\cdot)$. Thủ tục này được gọi là phương pháp lan truyền ngược sai số. Đầu tiên tính toán gradient của $Q(e(w,m))$ chỉ đối với vec tơ trọng số liên kết $w^{(\mu)}$ tại lớp μ, bỏ qua các lớp khác. Như vậy sẽ có:

$$\frac{\partial Q}{\partial w^{(\mu)}}=\left(\left(\frac{\partial Q}{\partial w^{(\mu,1)}}\right)^T,\left(\frac{\partial Q}{\partial w^{(\mu,2)}}\right)^T,\ \dots\ ,\left(\frac{\partial Q}{\partial w^{(\mu,N_\mu)}}\right)^T\right)^T. \tag{3.5.16}$$

Do hàm $Q(e(w,m))$ phụ thuộc vào vec tơ $w^{(\mu,i)}$ qua hàm vô hướng $y^{(\mu,i)}$ của (3.5.4) nên biểu thức (3.5.16) có thể viết lại như sau:

$$\frac{\partial Q}{\partial w^{(\mu)}}=\left(\left(\frac{\partial Q}{\partial y^{(\mu,1)}}\cdot\frac{\partial y^{(\mu,1)}}{\partial net^{(\mu,1)}}\cdot\frac{\partial net^{(\mu,1)}}{\partial w^{(\mu,1)}}\right)^T,\dots,\left(\frac{\partial Q}{\partial y^{(\mu,N_\mu)}}\cdot\frac{\partial y^{(\mu,N_\mu)}}{\partial net^{(\mu,N_\mu)}}\cdot\frac{\partial net^{(\mu,N_\mu)}}{\partial w^{(\mu,N_\mu)}}\right)^T\right)^T \tag{3.5.17}$$

Sử dụng (3.5.4) và (3.5.1) có thể viết nhân tử thứ hai và thứ ba của (3.5.17) như sau:

$$\frac{\partial y^{(\mu,i)}}{\partial net^{(\mu,i)}}=f(net^{(\mu,i)}) \tag{3.5.18a}$$

$$\frac{\partial net^{(\mu,i)}}{\partial w^{(\mu,i)}}=u^{(\mu,i)}. \tag{3.5.18b}$$

Nhân tử đầu tiên của (3.5.17) là hàm nhậy. Thành phần của tiêu chuẩn chất lượng tức thời $Q(e(w,m))$ đối với vec tơ đầu ra $y^{(\mu)}$ tại lớp μ.

$$S^{(\mu)}=\frac{\partial Q}{\partial y^{(\mu)}}=\left(\frac{\partial Q}{\partial y^{(\mu,1)}},\frac{\partial Q}{\partial y^{(\mu,2)}},\dots,\frac{\partial Q}{\partial y^{(\mu,N_\mu)}}\right)^T=(S^{(\mu,1)},\ \dots\ ,S^{(\mu,N_\mu)})^T. \tag{3.5.19}$$

Sử dụng quy tắc lấy vi phân hàm phức tạp và trên cơ sở cấu trúc mạng nơron nhiều lớp suy ra:

$$S^{(\mu,i)}=\frac{\partial Q}{\partial y^{(\mu,i)}}=\sum_{j=1}^{N_{\mu+1}}\frac{\partial Q}{\partial y^{(\mu+1,j)}}\cdot\frac{\partial y^{(\mu+1,j)}}{\partial y^{(\mu,j)}}=\sum_{j=1}^{N_{\mu+1}}S^{(\mu+1,j)}d_{i,j}^{(\mu+1)},\ i=1,2,\ \dots\ ,N_i, \tag{3.5.20}$$

Biểu thức (3.5.20) biểu diễn phương trình hồi quy đối với vec tơ thành phần độ nhậy $S^{(\mu)}$. Dạng ma trận của (3.5.20) sẽ là:

$$S^{(\mu)}=D^{(\mu+1)}S^{(\mu+1)},\ \mu=M-1,M-2,\ \dots\ ,2,1. \tag{3.5.21}$$

ở đây $D^{(\mu+1)}$ là ma trận chuyển độ nhậy có chiều $N_\mu \times N_{\mu+1}$ đối với mạng bậc nhất (3.5.1) và có dạng sau đây sau khi kết hợp với (3.5.8):

$$D^{(\mu+1)}=\frac{\partial y^{(\mu+1,j)}}{\partial y^{(\mu,i)}}=(\ d_{i,j}^{(\mu+1)}\ )=\frac{\partial y^{(\mu+1)T}}{\partial y^{(\mu)}}=W_1^{(\mu+1)T}\,diag(f^*(net^{(\mu+1)})) \tag{3.5.22}$$

Trong biểu thức (3.5.22) đối với ma trận chuyển độ nhậy $D^{(\mu+1)}$ lưu ý rằng ma trận $W_1^{(\mu+1)T}$ được xác định bằng (3.5.6)

$$f^*(net^{(\mu+1)})=[f^*(net^{(\mu+1,1)}),\ \dots\ ,f^*(\ net^{(\mu+1,N_\mu+1)})]^T.$$

Như vậy biểu thức (3.5.22) chỉ bao gồm phần ma trận trọng số liên kết $W^{(\mu)}$ tại các lớp μ không chứa vec tơ ngưỡng $w_0^{(\mu)}$.

Biểu thức gradient của $Q(e(w.m))$ đối với vec tơ $w^{(\mu)}$ trọng **số liên kết tại lớp μ** (3.5.17) có dạng:

$$\frac{\partial Q}{\partial w^{(\mu)}}=(S^{(\mu,1)}f^{*}(net^{(\mu,1)})[u^{(\mu)}]^{T},\ \ldots\ ,S^{(\mu,N_{\mu})}f^{*}(net^{(\mu,N_{\mu})})[u^{(\mu)}]^{T})^{T}. \quad (3.5.23)$$

Biểu thức (3.5.23 và (3.5.21) tạo thành quy trình tính toán tìm gradient phiếm hàm tức thời Q theo toàn bộ các lớp của mạng, bắt đầu từ lớp cuối cùng:

$$\frac{\partial Q}{\partial w^{(\mu)}}=\ \mathrm{diag}(f^{*}(net^{(\mu)}))S^{(\mu)}\otimes u^{(\mu)}. \quad (3.5.24a)$$

$$S^{(\mu)}=D^{(\mu+1)}S^{(\mu+1)}, \qquad \mu=M-1,M-2,\ \ldots\ ,2,1, \quad (3.5.24b)$$

trong đó $\otimes$ là ký hiệu tích trực tiếp của các ma trận hay còn gọi là tích Kroneker [41]

Ví dụ tích trực tiếp của 2 ma trận A và B là:

$$A\otimes B=\begin{Vmatrix} a_{11}B & \cdots & a_{1m}B \\ \vdots & \ddots & \vdots \\ a_{n1}B & \cdots & a_{nm}B \end{Vmatrix} \quad (3.5.25)$$

Các điều kiện ban đầu của (3.5.21) dễ dàng tính được theo công thức:

$$S^{(M)}=\frac{\partial Q(y^{(M)}-y^{*})}{\partial y^{(M)}}$$

và như vậy $S^{(M)}$ phụ thuộc vào cấu trúc hàm Q. Nếu Q có dạng bình phương như ở (3.5.14) và $R=0.5I$ với I là ma trận đơn vị thì vec tơ ban đầu $S^{(M)}=y^{(M)}-y^{*}=e$. Đây chính là lý do để giải thích xuất xứ của phương pháp lan truyền ngược sai số. Có nghĩa là quy trình (3.5.21) tính các sai số ban đầu của mạng theo hướng giảm dần số lớp, bắt đầu từ lớp M - lớp ra của mạng.

4 Tổng kết

Mạng nơron là một công cụ hữu hiệu cho các bài toán nhận dạng mô hình và điều khiển phi tuyến các hệ động học có độ bất định cao. Lớp mạng quan trọng ứng dụng trong nhận dạng mô hình và điều khiển là mạng RBF hoặc mạng nhiều lớp với thuật học lan truyền ngược sai số. Tuy nhiên thuật học này khó có thể áp dụng cho mạng nơron nhiều lớp cỡ lớn. Cần lưu ý rằng các tín hiệu trong mạng nơron sinh học chỉ có thể lan truyền theo hướng thẳng. Một số thuật học tự tổ chức hoặc thuật di truyền có thể sử dụng trong các bài toán điều khiển thích nghi thông minh và hứa hẹn nhiều ứng dụng có ý nghĩa. Mạng nơron còn có thể kết hợp với logic mờ tạo thành một hướng ứng dụng khá mạnh cho các bài toán nhận dạng mô hình và điều khiển trong công nghiệp

Tài liệu tham khảo

[1] **P.J Autsaklis and P.M Passino**: Introduction to intelligence control systems with high degree of autonomy. Kluwer. 1992.

[2] **S. Akhyar and S. Omatu**: Neuromorphic self-turning PID controller. IEEE Int. Conf. Neural networks, San. Francisco, March 28-Apr.1, Vol 1. 1993.

[3] **S. A. Billings**: Practical Indentification of NARMAX Models using RBF. Inter. J.Contr. Vol 52, p.1327-1350, 1992.

[4] **S. A Billings and S. Chen**: Properties of neural networks with application to modeling non-linear dynamical systems. Int. J.Contr., Vol 55, N1, p. 193-224,1992.

[5] **S. Chen**: Orthogonal least squares learning algorithm for RBFN, IEEE Tr. N.N, N2, p. 302-309. 1991.

[6] **S. Chen and S. A. Billings**: Neural networks for nonlinear dynamic systems modelling and identification. Int. J. Contr.. Vol 56, p. 319-349, 1993.

[7] **X. Chen and F. Q. Gao**: Adaptive control based on RBF networks, Proc. of 35 th Conf. on Dec. and Contr., p.3810-3816, 1996.

[8] **L. O. Chuna and L. Yang**: Celluler neural networks. IEEE Tran. Circuits and systems theory, p.1257-1290, 1988.

[9] **C. F. N. Cowan, S. A. Billings**: Nonlinear systems identification using RBF, Int. J.Contr., Vol 21, p.2513-1533, 1990.

[10] **R. J.Craddock, K. Warwick**: Multilayer Radical Basis Function networks and Extention to the radical basis function. IEEE Inter. Conf. on Neural Net., p. 700-705, Washington D.C 1996.

[11] **D. Flynn**: Neural Control of turbogenerator systems, Automatica, Vol 33, N11, p.1861-1873, 1997.

[12] **L. Fortuna and S. Geaziani**: Improving back-propagation learning using auxiliary neural networks. Int. J. Contr., V55, N4, p.793-807, 1992.

[13] **K. Godfrey, R. Jones**: Signal processing for control, Springer-Verlag Press, 1986.

[14] **Y. Hirose and K. Yamashita**: Back-propagation algorithm which varies the number of hidden units. Neural networks with application to modelling non-linear dynamical systems, Int. J.Contr., V55, N1, p.193-224, 1992.

[15] **K. J. Hunt, D. Sbarbaro, R. Zbikowski and P. J. Gawthrop**: Neural networks for Control systems: A survey, Automatica, V28, N6, p.1083-1112, 1992.

[16] **K. J. Hunt, G. Irwin**: Neural Network Engineering in dynamic Control Systems, Springer-Verlag London Limited. 1996.

[17] **M. Kawato**: Computational schemes and neural network models for formation and control of Multijoint Arm Trajectory Neural Networks for Control, MIT Press, 1990.

[18] **A. N. Kolmogorov**: On the representation of the continuous functions of the several variables by superposition of continuous function of the one variable and addition. Dokl. Acad. Nauk USSR. Vol 114, p.953-956, 1957.

[19] **B. Kosko**: Neural Network and Fuzzy systems. Prentice Hall, Englewood Cliffs, 1992.

[20] **K. Kristinsson and G. A. Dumont**: System Identification and control using genetic algorithm, IEEE TR. Syst. Men and Syber. 22(8), p.1033-1046, 1992.

[21] **C.T. Lin and C. S. G. Lee**: Neural Fuzzy systems. Prentice Hall, 1996.

[22] **W.T. Miller**: Rapid learning using CMAC neural Networks: Real_Time control of unstable system, Proc. of IEEE Intelligent Control Symposium, Philadelphia, 1990.

[23] **K. S. Narendra and K. Parthasarathy**: Identification and Control of dynamical systems using Neural Networks, IEEE Tr. N.N. V1, N1, p.11-27. 1990.

[24] **K. S. Narendra and S. Mukhopadhyay**: Intelligent Control using neural networks, IEEE Control System, p.11-18, 1992.

[25] **H. D. Nguyen, B. Window**: Neural Network for self-learning Control Systems. IEEE Control System Magazine. Vol 10, N3, p.18-23,1990.

[26] **X. Ni and S. T. R. Simons:** Nonlinear Dynamic System Identification using RBF networks, Proc. of the 35 th Conf. On Decision and Control, p.935-937, 1996.

[27] **F. M. Pottman and D. E. Seborg:** A nonlinear prediction control strategy using RBF Networks, IFAC Symposium DYCORD, College Park, p. 309-314, 1992.

[28] **G. Puskorius and L. Feldkamp:** Neuro-control of nonlinear dynamical systems with Kalman filter trained recurrent networks, IEEE Tr. NN, Vol 5, N2, p.279-297, 1994.

[29] **S. Z. Qin and H. T. Su:** Comparison of 4 Neural net learning methods for Dynamical System Identification, IEEE Tr. N.N. Vol 3, N1, p. 112-130, 1992.

[30] **D. E. Rumelhart, J. I. McCellawd Eds:** V1, 2, cambridge, MA, MIT Press, 1986.

[31] **D. E. Rumelhart, G. E. Hinton, R. J. Willioms:** Learning internal reprentation by error propagation, Paralled Distributed Processing, Cambridge, MA MIT, Press V1, 1986, p.318-362.

[32] **R. M. Sanner and J. E. Slotine:** Stable adaptive control and recursive identification using radical Gaussian networks. Proc. of the 30 th conf on decision and control, Brighton, England, 1991.

[33] **M. A. Sartori and P. J. Antsaklis:** A Gaussian Neural Network Implementation for Control Scheduling, Proc. of the IEEE Int. Sym. On Intell. Control, p.400-404, 1991.

[34] **M. A. Sartor:** Feedforward neural networks and their applications in the Hight level control of systems, PhD Dissertation, University of Notre Dame, 1991.

[35] **R. Schalkoff:** Artificial neural networks, Mc Graw-Hill, International Editions, 1997.

[36] **M. H. Stone:** The generalized Weierstrass approximation theorem, Mathematica Magazine, Vol. 21, p.167-184, 1948.

[37] **R. S. Sutton and A. G. Barto:** Reinforcement learning in direct adaptive optimal control. Control system Magazine, Vol 12, N3, 1992.

[38] **Ya. Z. Tsypkin:** Adaptation and learning in Automatic Systems, Academic Press., NewYork, 1971

[39] **S. V. Ulyanov and V.N. Zakharov:** Towards design of AI Control systems. Proc. East-West Conf. Art Intelligence: From theory to practice, EWAIC 93, Moscow, Sept. 7-9, 1993.

[40] **K. Warwick:** An overview of neural networks in control applications, In "Neural networks in Robotic control theory and applications", Ellis Horwood, 1995.

[41] **A. Weimann:** Uncertain models and Robust control, NewYork, Springer-Verlag, 1991.

[42] **P. J. Werbos:** Beyond regression: New tools for prediction and analys in the behavioral sciences. Thesis in applied mathematics, Harvard University, 1974.

[43] **W. J. Willis, G. A. Montague, C. D. I. Massimo, M.T Them and A.J. Morris:** Artificial Neural Networks in process Estimation and Control. Automatica, Vol 28, V 6, p.1181-1187, 1992.

[44] **T. Yabuta:** Neural Network controller characteristics with regard for adaptive control, IEEE Tr. Syst. Man and Cyber., V22, N1, 1992.



1956 - 2006

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
50 NĂM XÂY DỰNG VÀ PHÁT TRIỂN

206203



Giá: 40.000đ